Plans Unfolding 1.1
For Letter/A4

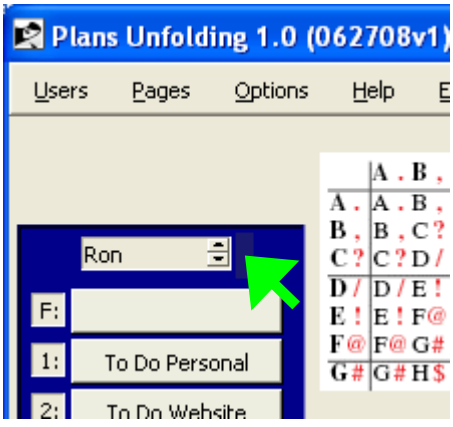
Users Manual

Ron Doerfler 10/05/2008
http://www.plansunfolding.com/

NOTE:

IF YOU CANNOT GET THE CIPHER PAGE BUTTONS TO WORK:

For added protection of your keys, you **must** click on a small, almost imperceptibly darker rectangle just to the right of your user name in the upper spinbox after you check the **Show Keys in PlainText** button or click on the **Synchronize** button to make them work.



Contents

Contents 2
Acknowledgments 3
Release Changes 3
  Version 1.0 to Version 1.1 3
Introduction and Overview 4
Installation and Quick Start 5
  Upgrading a Previously-Installed Version of the Same Letter/A4 Type 5
  Initial Installation or Switching Between Letter/A4 Versions 5
  Quick Start 6
  Printing and Folding 6
    Folding the organizer into a booklet 7
    Paging through the booklet 8
    Printing and Folding When There Are Large Printer Margins 9
Using Your Organizer 10
  Menu Options 10
  If the Interface Software Hangs When Loading a Page 11
  Organizer Pages 12
    List Page Type 14
    2-Field,1-Column Page Type 15
    2-Field,2-Column Page Type 16
    Calendar Page Type 17
    Cipher Table and Cipher Text Page Types 18
    Treasure Map Page Type 22
    Astronomy Page Type 22
    Astronomy Page Type 23
    Text Page Type 24
Customizing Your Organizer 30
  Vertical Tables 30
  Horizontal Tables 34
  Writing Page Scripts that Work on Letter and A4, Middle and Corner Pages 37
  Lines and Repetition 40
  Mixing Text Nodes and Shapes 43
  Variables: Counters and Lengths 46
  Creating Custom Commands 47
  Creating Custom Colors 48
  Overlaying Images 48
  Changing Fonts 48
  Sharing Page Scripts 50
Support 50
Copyright Notice 50
Appendix A: LaTeX Add-On Packages 51
Appendix B: Manual Installation of MiKTeX Packages 52
Appendix C: Updating MiKTeX and Its Added Packages 52
Appendix D: Absolute Page Coordinates 53
Appendix E: Structure of the Code 54
Appendix F: Troubleshooting 58

Acknowledgments

My sincere thanks goes to the following people who helped make this product possible: Brian Doerfler, who started me on this paper organizer kick a few years ago and has helped with the design and test of it throughout; Derek Doerfler, who created the online forum and galleries to support users of the program; Stephen L. Moshier (http://www.moshier.net), who wrote the aa-56 program used to compute ephemerides of solar system bodies for creating the Astronomy page; Keith Burnett, whose data files of star positions and magnitudes, constellation figure lines and constellation labels (links are listed on http://bodmas.org/kepler/horizon.html) were also used in creating the Astronomy page; and Melissa Hall, who created the Treasure Map graphic and made it available for use here.

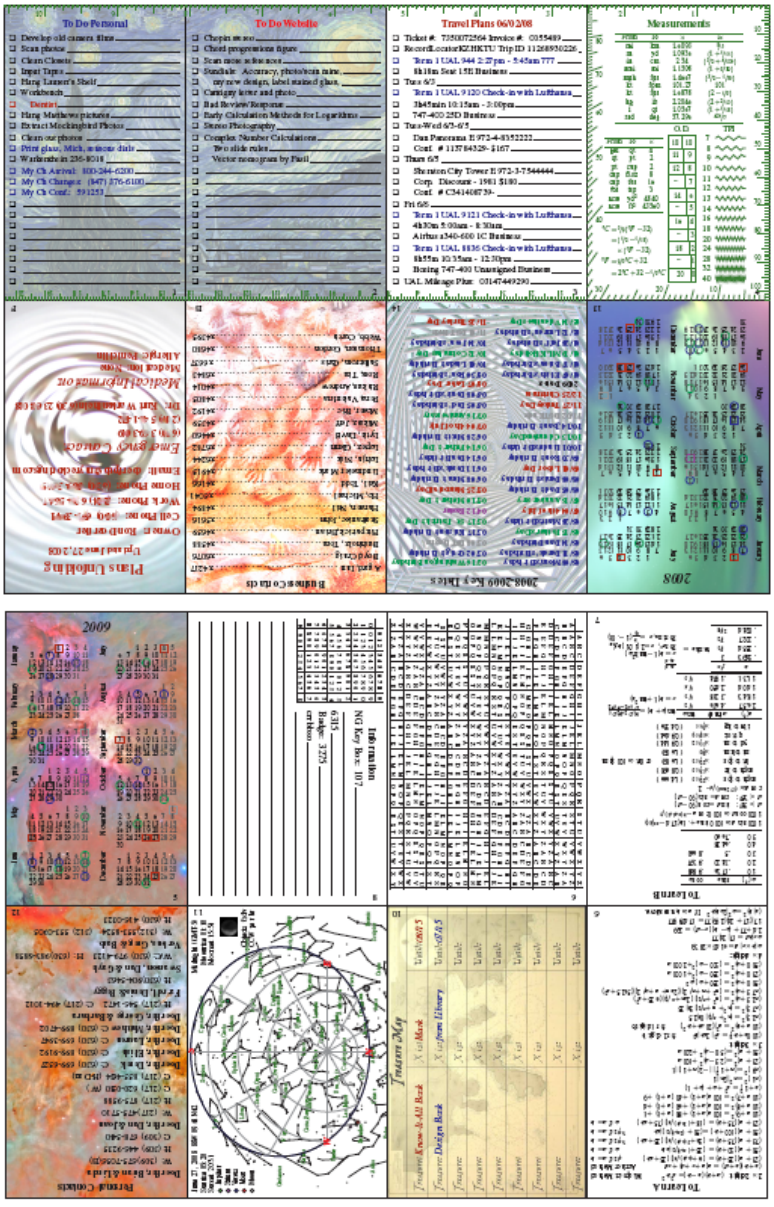
Release Changes

Version 1.0 to Version 1.1

- Software
  a. Added default fonts and font sizes to fix oversize widgets if user has large default Windows fonts.
  b. Added a menu option to change user default fonts and font sizes.
  c. Added a menu option to copy a user to a new user.
  d. Added menu options to turn off printing of fold lines and/or page numbers.
  e. Added preset variables for page location (corner or middle) and full and indented page widths as each page is loaded so that page scripts can accommodate different display widths.
  f. The executable file is now simply named *PlansUnfolding.exe* (no version number in name).

- Users Manual
  1. Added text throughout the manual relevant for the new A4 paper size version of the software.
  2. Added instructions for upgrading to Version 1.1 from an existing Version 1.0. Note that the executable file is now named *PlansUnfolding.exe*, so please update any desktop shortcuts to this new file and verify that the titlebar of the window says "Plans Unfolding 1.1" when you launch the program.
  3. Added instructions for using A4 paper and for printers that don't print far enough to the edge.
  4. Added a description of the menu option to change fonts and font sizes.
  5. Added a description of menu option to copy a user to a new user.
  6. Added a description of menu options to turn off printing of fold lines and/or page numbers.
  7. Added a caution that when calculating coordinates the value with units must come first.
  8. Added a description of use of \providecounter and \providelength to declare counters and lengths that can be used in multiple page scripts.
  9. Added a description of creating custom commands for use in scripts.
  10. Added examples of writing page scripts that work for both middle and corner page widths.
  11. Added a description of how page scripts can access variables specific to the current page (location, width, indented width, doubly-indented width, etc.).
  12. Added an example of a simple \foreach loop in a page script (music staffs) that also demonstrates the use of the \global\advance command for incrementing lengths within \foreach loops.

Introduction and Overview



Welcome to the world of the **custom color paper organizer**! With today's access to color printers you can print and fold a two-sided unique planner designed by you. Assign standard page types to any page, or download or create your own page designs. Use whatever images you want as color backgrounds—family, pets, fine art, or any of your interests. Or just start right away entering your todo's, dates and contacts using the default set of pages—it's really very easy to use. And best of all it's freeware.

The *Plans Unfolding* software provides a complete, convenient user interface for Windows to arrange and maintain your data. It is backed by the powerful LaTeX typesetting engine coupled with the TikZ vector drawing package to create professional-quality letter-size output. With a few folds and a single scissor cut you have your own custom 16-page pocket planner and organizer. Cross off entries and add new ones as you go. Print out a new one whenever it's convenient for you.

Multiple users are supported—perhaps you want one organizer layout for trips and one for general use, or more than one person in the household is interested in using it. All data and all preference settings are kept separate, and switching between users is as simple as clicking on a button.

The *Plans Unfolding* home page is at http://www.plansunfolding.com. A *Plans Unfolding* online forum for sharing ideas, suggestions and questions can be found at http://www.myreckonings.com/forum. Links to a Background Image Gallery and a Custom Page Script Gallery can be found under the *Plans Unfolding* forum—download shared images and custom page designs or upload your own to the galleries. One of the really special things about this organizer is its ability to share and edit custom graphic page designs by pasting simple text (a "page script") into that page's entry field in the interface.

At this point *Plans Unfolding* runs only on the Windows operating system (2000,XP,Vista). There is a version for letter-size paper and another for A4 paper, both sharing this Users Manual. I am pursuing a version that will run on Linux, Mac OSX and other Unix-based systems. Please see the *Plans Unfolding* home page at http://www.plansunfolding.com for the latest news on this.

Use the *Plans Unfolding* organizer to record expenses; track mileage; list travel information, create shopping and grocery lists; list medications and times; write useful formulas and conversions; maintain and check off todo lists, jot things to remember; mark important dates; list game schedules, bets and scores; reference video poker odds; enter gift ideas and clothing sizes; encrypt private information using the built-in cipher page, and so on!

Installation and Quick Start

Plans Unfolding uses the LaTeX typesetting language to produce professional-quality organizer output. Therefore, you must be connected to the Internet to download the LaTeX distribution and the required add-on packages. The initial installation process takes a good deal of time but the results of using LaTeX are worth it. This application has been extensively tested with the free MiKTeX distribution of LaTeX, so I strongly urge you to use this distribution. The "Basic" MiKTeX distribution requires 255MB of disk space when installed (plus 77MB of download), so don't download the "Complete" distribution unless you really want it for different uses.

Upgrading a Previously-Installed Version of the Same Letter/A4 Type

Installing a new version of *Plans Unfolding* over an existing version is very quick. There is no need to re-install the MikTeX distribution of LaTeX. **Do not delete any files in the directory where the existing version resides**, as some of these files contain your user data that you want to retain.

1. Copy the folder where you previously installed *Plans Unfolding* to a temporary folder as a backup.
2. Open the zip file of the new release and extract **all** the files into a second temporary folder.
3. Copy all the unzipped files from this second folder into the original folder where you installed *Plans Unfolding*, choosing to "**Yes To All**" when asked whether to overwrite files in the original folder.
4. Starting with Version 1.1 the executable file name is named *PlansUnfolding.exe*, so any desktop shortcut linking the old *PlansUnfolding_10.exe* filename must be updated to this new filename.
5. Launch *PlansUnfolding.exe* in the installation folder. Please verify that the correct new version number appears in the title bar of the window.
6. If there are any additional packages required in the new version, MikTeX should prompt you to have them automatically installed—otherwise, you may have to install it using the Package Manager described in Appendix B. Version 1.1 requires **no** additional packages.
7. After verifying that the new release works, you can delete the two temporary folders from Steps 1 and 2.

Initial Installation or Switching Between Letter/A4 Versions

1. **Download and install the MiKTeX distribution of LaTeX (NOT REQUIRED if simply switching versions!)**. Go to http://www.miktex.org and click on the latest MiKTeX release (MiKTeX 2.7 assumed here) in the Download section of the left frame. Download and install the "Basic" MiKTeX system. Follow the install prompts, and when you get to the Settings window select "Letter" or "A4" for the preferred paper and "Ask me first" for installing missing packages. The installation will take some time—don't worry if it seems to stall.

2. **Unzip the *Plans Unfolding* zip file into any convenient directory if you haven't done it already** (this manual is available separately so you may not have unzipped it yet). No windows installation is required.

3. **Automatically install additional packages required by *Plans Unfolding*.** Double-click on the *Plans Unfolding* executable file (named *PlansUnfolding.exe*). Enter a user name at the prompt and click OK, then re-launch the application to load the initial set of default user pages. Then click on the **Save and Build** button to run an initial build. You should be prompted to download and install the following additional packages one at a time if they have not already been downloaded when running a previous Letter/A4 version:

   memoir symbol was xcolor preprint units faktor microtype colortbl
   ms pgf xifthen forloop datatool xfor fp substr lettrine

   Accept each install and wait for the black pop-up window to disappear. After the first one or two downloads, you can uncheck the checkbox in the prompt window so all subsequent package installations are done without stopping to ask. Not all of these additional packages are required today—they represent a useful set that can be leveraged in future releases or by you if you customize your pages.

4. **If you are not prompted, or if the package download terminates before you see a final confirmation message**, you may have to install the packages through the MiKTeX Package Browser. See Appendix B for instructions if you need them.

The one-time installation is now complete and the application is now ready for use.

5. **At some point after evaluating *Plans Unfolding*, you may want to install any MiKTeX updates**. See Appendix C for instructions.

Quick Start

1. **Double-click on the *Plans Unfolding* executable file** if it is not running already.

2. ***DIVE IN* and edit the pages as you wish**. The calendars are pre-populated with common U.S holidays and there is sample text on most pages as hints. If you want to change any page type, use the Pages→Change Page Types menu command (for example, you can have lists for all 16 pages if you want!). Any background images you select must be stored in the BackgroundImages subdirectory of the main *Plans Unfolding* directory.

3. **Save your pages to the user Recovery Point** once you have something you want to save as a backup. Changes to individual pages or to all pages since the current launch of the application can be undone via the buttons along the left side of the interface. Saving to a Recovery Point after a successful build is **highly** recommended.

4. **Click on the "Save and Build" button**. After 20 seconds or so of processing, the organizer PDF file will appear (if the application returns control without the PDF file appearing, review the build output text for any errors—otherwise the PDF file should be found in the subdirectory named after the user.)

5. **Print and fold the organizer**. See the following *Printing and Folding* section.

Printing and Folding

You don't have to have a color printer for this, but it does make a world of difference in how the organizer looks and how easy it is to use. Better paper designed for color printouts gives a really professional look to the organizer. Colored paper or paper with pre-printed color background designs or marbling will really help a black-and-white printout for this. You may find that your printer will not print to the ¼" border of the organizer; if no content is lost this border loss is not important. **The instructions here are valid for either letter-size paper or A4 paper**, and following this is advice for cases where your printer margins are large enough to chop off parts of the corner pages.

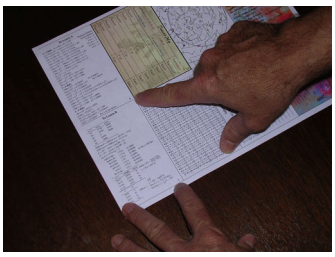
You will want to use the following printer settings when your printer can print far enough to the edge to read content in all pages:

- Landscape orientation.
- Two-sided without the "flip up" option. You can use a single-sided printer by printing one side, then re-inserting the paper to print on the other side such that pages F and 1 are opposite pages 6 and 7 when you're done.
- **No** page scaling or fitting to printer margins!
- **No** centering or auto-rotation!

You may find that your printer does not align the middle vertical fold line so that the first fold is right along that line. There is an Adjust Printer Margins option under the Options menu where you can add +/- offsets to the left margins of each side in points (1mm = 2.85pt), which can get that fold line perfectly in the middle. It turns out that printer misalignment only becomes noticeable at all in the folded booklet when you get up to about 1mm error.

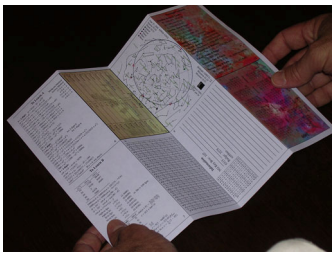
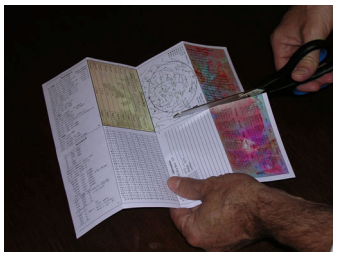
Folding the organizer into a booklet

It's actually fast and easy to fold the organizer into a booklet that provides ready access to all 16 pages. The photos below probably make it look more difficult than it really is, but I wanted to be very clear on it.

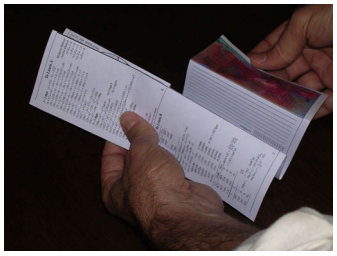
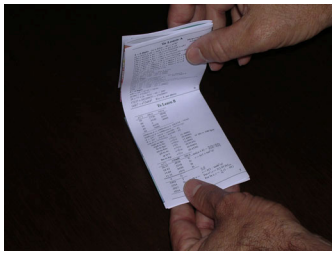


Place the paper with pages 6 and 7 face up and located on the left side of the sheet (the finger is pointing to the number "6" on that page. Make a valley fold in the middle (parallel to short sides of paper) and crease.

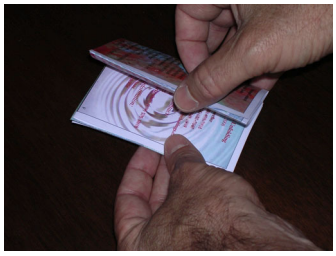
Make a mountain fold along the outer pages on each sides to fold them backwards over the first fold.

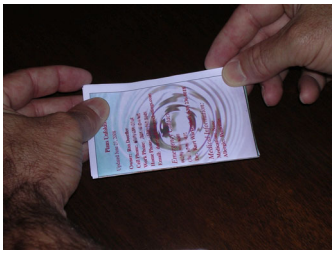
Cut along the long middle line from the right edge but do not cut through the line separating page 6 from page 7.

Collapse the fan fold for the top and bottom halves behind pages 6 and 7, then hold the ends of pages 6 and 7 and fold page 6 (with its fan fold) over page 7.

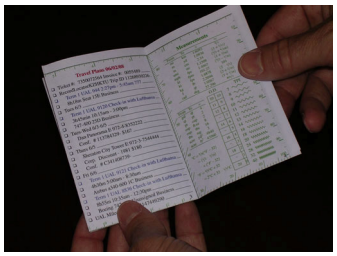


Crease this new fold between pages 6 and 7, then open the organizer at pages F and B and fold B behind F.

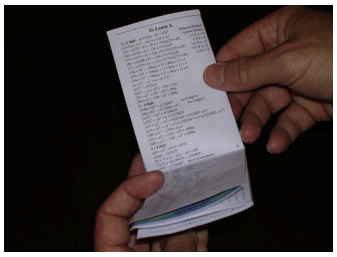
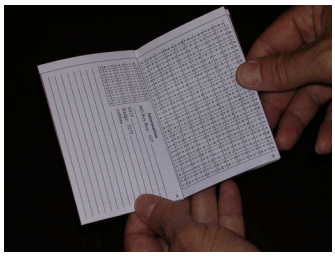


Align and hold the outer corners as you crisp up the left fold to align all edges. The organizer is finished!

Paging through the booklet



All 16 pages are available for use as a little booklet. Open up the first page to view pages 1 and 2, then the next page to view pages 3 and 4. (You can pull out the fan fold at page 4 to use the rulers.) Repeat to view pages 5 and 12.

Fold page 12 backwards behind the stack of pages (to be viewed later). Open the booklet vertically to view pages 6 and 7 as one doubly-long page, then reverse that to return to page 5. Flip page 5 to view pages 6 and 7.

Open the next (doubly-thick) page to view pages 10 and 11. Flip page 11 over to see page 12. Flip page 12 over to see pages 13 and 14.

Flip page 14 over to view the back page B and fold it back behind page F and you have returned to the beginning of the booklet. Every page was viewed with only one cut and no stapling at all! And there's even a pocket behind the the front page that can hold business cards.

If you plan on carrying the organizer in your pocket for a long time between updating and printing a new one, you might stick some clear tape between pages B/F on the left edge and and F/2 along the bottom edge. I personally use a large-size soft-vinyl badge holder to hold the organizer in my back pocket. This works great for **letter-size paper** and lasts a long time. After much searching I found the absolutely perfect holder to be Baumgarten's "BAU 67880" clear vertical badge holder. Googling the text above in quotes will produce online suppliers—unfortunately, they come in packs of 50, but at about $10.60 per pack plus shipping.

Printing and Folding When There Are Large Printer Margins

Some printers simply won't print to within 1/4" of the edge of the paper where the organizer border lies. If this cuts off the border and a bit of blank space in the edge pages, this is not a big deal. However, if you are losing printed content on the sides, and you have already tried the printer settings above that do **not** re-scale or auto-center the output, then there is a way to print out the whole organizer but just slightly smaller in size.

First, when printing set your printer dialogue to "Fit to Printer Margins" or "Shrink to Printer Margins," which will shrink the PDF output the same percentage in each direction to fit within the narrowest margin on your printer. Then the printed PDF file will have extra margins. If you have a single-sided printer, the printer margin adjustment for one side can help with any mismatch in fold lines between sides.

There will be extra white space, probably on all sides of the printout. So start the folding process, doing the valley fold and the first two folds back on the valley fold, creasing on the fold lines rather than lining up the edges of the sheet. Here the extra margins on the outsides are clearly seen. You can lift each flap and cut the excess off of each edge with a scissors by simply using the correct edge of the original valley fold as a guide to cut along. Then continue the folding all the way to the end. When you're done you can trim some of the excess white space off the top of the final booklet, and you end up with just a slightly smaller organizer--you couldn't tell it's smaller unless

you saw another one, and it may actually be a bit more convenient in size for you.

This trimming is very fast and the booklet looks just fine. However, the rulers will not be scaled correctly so you should disable printing them with the "Options-->General Settings" menu function.

Using Your Organizer

Plans Unfolding offers a convenient user interface for entering and maintaining planning and reference information on 16 mini-pages that can be printed on two sides of a letter-size or A4 sheet of paper. As described in the previous section, when the printout is folded the planner becomes a small, thin booklet that is easily carried in a pocket. On the fly updates can be penciled in and then added to the organizer at the next launch of the interface software.

The default page type distribution was designed for quick reference, but you can change the page types to any of the supported types or any custom page type. The default page types have a convenient Business Contacts listing on the back so it's viewable even through a plastic sleeve. Open the next page from the back and you see pages 13 and 14, containing the calendar for the current year alongside a listing of Key Dates for this year color-coded with tags in the calendar (because the tagged calendar dates were exported to the Key Dates page as described below). Going forward from the front page F, we open to list pages 1 and 2, then list page 3 and a measurements page 4. Pulling the fan-fold outward between pages 4 and 1 extends the rulers across the entire width of Side 1 of the organizer for easy measuring. Pages 6 and 7 fold downward for one doubly-long page for long content spread across them (at one point I had blank music staffs running down these two). The more complicated pages (cipher, astronomy if selected, etc.) are in the interior pages. So there is some method to the seemingly random default assignment of page types between the pages, but again you can change all of this.

We will discuss the general menu options available in the software interface, followed by all the standard interface-supported page types supported by the interface. We will end this section by describing some basic LaTeX commands to spruce up your Text pages before launching into the next section on creating custom page designs.

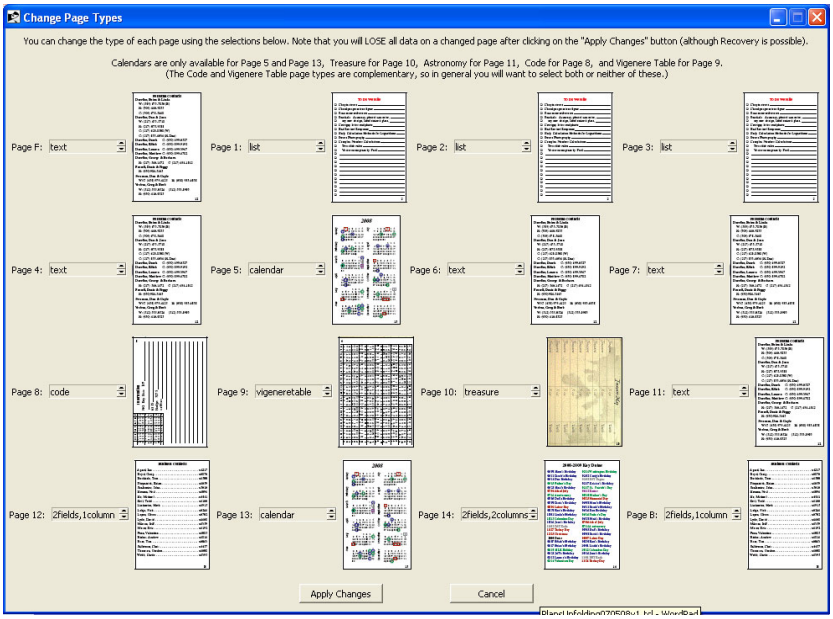
Menu Options

Here are the menu options available in the interface software:

Users→Add User	Create a new user with default page layout and sample entries in the various pages.
Users→Copy User to New User	Copy current user data to a new user (for testing new background image themes, creating a travel version, etc.)
Users→Delete User	Delete the entered user—there is no Undo for this function.
Pages→Save All to Recovery Point	Save all current pages to a Recovery Point (recommended!).
Pages→Recover All from Recovery Point	Recover all pages for a user from the Recovery Point.
Pages→Change Page Types	Change the type of any or all pages (see the figure below).
Options→Adjust Printer Margins	Adjust the left margin of each side for your printer. If the provided range is not sufficient, you will have to adjust the default margins within the printer interface window when you print the organizer (this is sometimes found under an "Advanced" tab or button).
Options→Change Interface Font	Change the default font and fontsize for the interface (not the printout). **Please make sure that your selected widget font size roughly centers the page buttons in the left blue panel as shown in screen captures here!**
Options→Add Custom LaTeX Code	Add LaTeX script to the Preamble or to Side 1 or Side 2.
Options→General Settings	Turn on/off protractor, rulers, fold lines, and page numbers., and

	also input user latitude, longitude and timezone if using the Astronomy page.
Help→User's Manual	Show where to find this manual—should launch it as well.
Help→About…	Show current software version and homepage URL.
Error Just Happened?-->Prepare Debug Files	Copy Recovery Point, Backup and current page scripts into the DebugFiles directory of the user, along with a session log file, etc. After carefully deleting any confidential information, you can zip these files and email them for help in analyzing the problem if you would like (see the *Support* section of this manual).

Shown below is the window used to change the type of each page to standard page types supported by the interface as discussed in the next section. The specialty pages (calendars, cipher table and cipher text, and astronomy) are restricted to certain pages as described in the text at the top of this window.


Change Page Types
You can change the type of each page using the selections below. Note that you will LOSE all data on a changed page after clicking on the "Apply Changes" button (although Recovery is possible).
Calendars are only available for Page 5 and Page 13, Treasure for Page 10, Astronomy for Page 11, Code for Page 8, and Vigenere Table for Page 9.
(The Code and Vigenere Table page types are complementary, so in general you will want to select both or neither of these.)
Page F: text
Page 1: list
Page 2: list
Page 3: list
Page 4: text
Page 5: calendar
Page 6: text
Page 7: text
Page 8: code
Page 9: vigeneretable
Page 10: treasure
Page 11: text
Page 12: 2fields,1column
Page 13: calendar
Page 14: 2fields,2columns
Page 8: 2fields,1column
Apply Changes
Cancel

If the Interface Software Hangs When Loading a Page

If the interface software hangs when it is trying to load a page, the page script has been corrupted or there are errors in a custom page script that make interpretation of the page contents impossible. In this case you cannot recover your pages from the Recovery Point via the menu because the software is non-operational. Enter Ctrl-Alt-Delete to pull up the Task Manager in Windows, select the *Plans Unfolding* application, and click on End Task to kill it. Then double-click on the *Recover_Pages_If_Application_Hangs.exe* file in the Plans Unfolding directory and enter the user name to recover their files. Then you can run the interface software again and it will use the recovered pages. See the *Support* section of this manual if you are still having problems.

Organizer Pages

The software interface directly supports several types of color page layouts via the interface:

- **List** – List for to-do items, ideas, travel data, etc.
- **Text** – Either simple text or text and tables formatted using LaTeX commands, for reference material, listings of contact information, etc.
- **2-Field,1-Column** – Matching pairs of data such as names and phone numbers, separated by dots
- **2-Field,2-Column** – Matching pairs of data in smaller font and two columns, separated by a space
- **Calendar** – Calendar with tagged dates selected by the user
- **Cipher Table and Cipher Text** – A Vigenere Cipher Table and a Cipher Text page for encrypted data
- **Treasure Map** – Location information for items that are lent, shared or placed somewhere
- **Astronomy** – Sky map and almanac data for a user location (North Hemisphere only)

These are described individually in the following sections. In the *Customizing Your Organizer* section we will see how pages can be assigned these page types and how custom pages can be created and shared online.

A newly-created user will have a default combination of these page types with sample text sprinkled through them. When the PDF output file is built, it will consist of two pages as shown on the next page (but with a few backgound images). We'll use these default pages to review the page types.

In this section any default backgrounds have been eliminated to make the page contents easier to view as smaller images. Keep in mind that background images can transform a dull organizer to something truly unique.

Background Images

Most pages have a **Background Image** field in which you can specify .JPG, .PNG or .PDF file to use as the page background. Please include this extension on your filename. These files must be located in the *BackgroundImages* folder under the main folder where you placed the *Plans Unfolding* files. These images will be automatically squeezed or stretched to match the page dimensions:

- Corner pages: 2.50" wide by 4.05" tall for letter; 69mm wide by 100m tall for A4 size paper
- Middle pages: 2.77" wide by 4.05" tall for letter; 74mm wide by 100mm tall for A4 size paper

Due to the high resolution of color printers today, I would suggest sizing images at 4x each dimension (10" or 11.08" wide by 16.20" tall for letter, 276mm or 296mm wide by 400mm tall for A4 paper) and letting the *Plans Unfolding* software reduce them, or at least ideally crop them to a 1:1.62 ratio of width to height for corner pages or 1:1.46 ratio for middle pages for letter, or else 1:1.45 ratio of width to height for corner pages or 1:1.35 ratio for middle pages for A4. Images are ***not*** rotated for horizontal pages in the application, so make them **all** vertical images—if you want a car to be horizontal on a page you are configuring for horizontal orientation, create an image for it in which the car is vertical. You can also use multi-image sets on pages that have the same vertical alignment to get one large image, such as the Van Gogh *Starry Night* painting seen in the *Introduction and Overview* section.

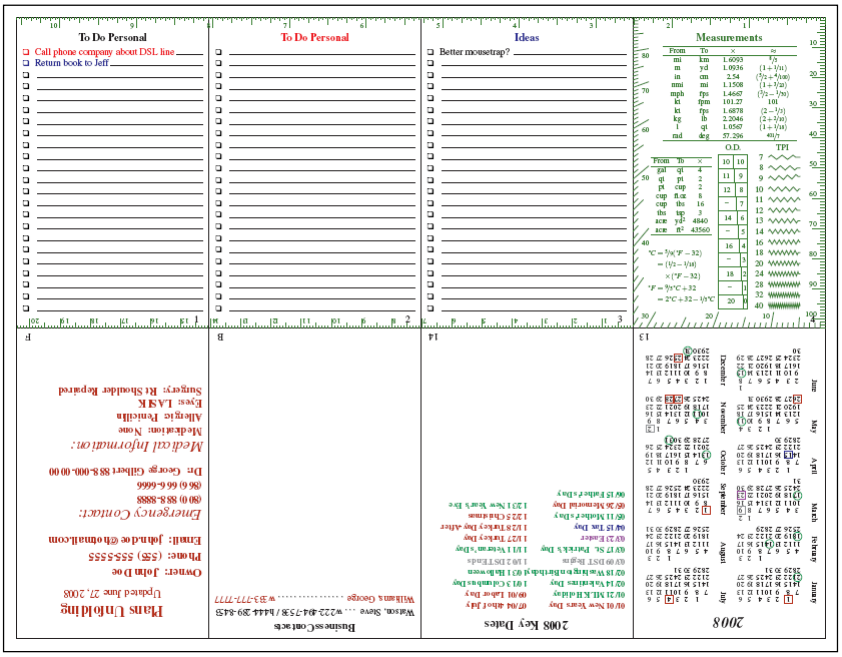
Stop by the Background Images Gallery online, a child board of the Plans Unfolding forum at http://www.myreckonings.com/forum to download or upload images in the public domain or with a Creative Commons license. Or use your own .JPG images of your kids, whole family, pets, car, etc. to really personalize your organizer! And remember, you can perform a high-resolution scan any type of reference material you want, such as graphs and other technical illustrations, save it as a .JPG or .PNG file, and use this file as a background image to print this data onto an organizer page in whole. Generally an image will look fine in either a corner or middle page.

Switching Users

Access to pages for different users is as easy as clicking on the spinbox in the upper left of the interface.

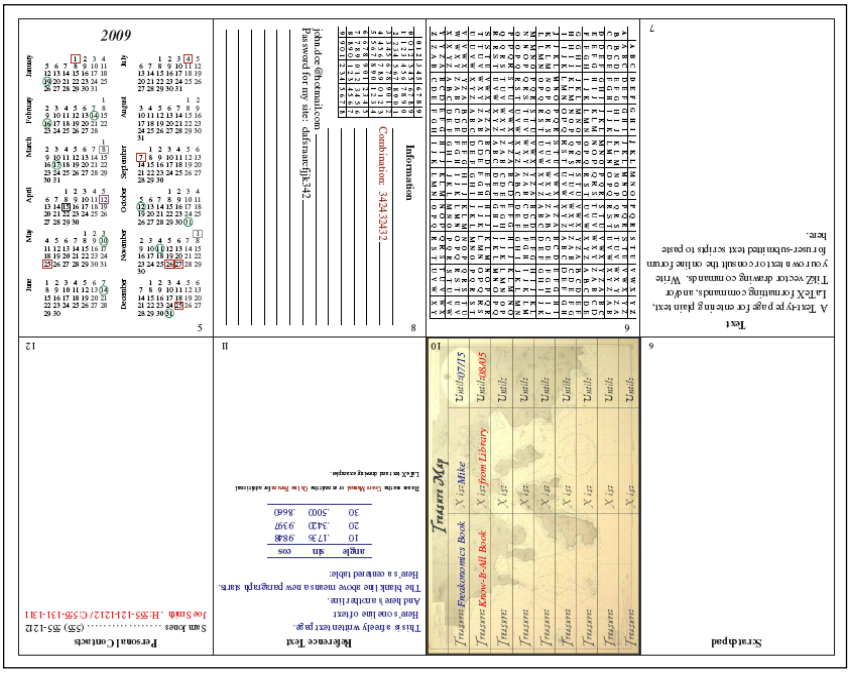
Side 1 of the default organizer consists of

- three List pages 1, 2 and 3
- a plain Text front page F
- a 2-Field,1-Column back page B
- a 2-Field,2-Column page 14
- a 2008 Calendar page 13
- a custom Text page 4 for the letter-size version, or a List page for the A4 version



Side 2 of the default organizer consists of

- a 2008 Calendar page 13
- a Vigenere Cipher Table page 9
- a Cipher Text page 8
- plain Text pages 6 and 7
- a 2009 Calendar page 5
- a custom Text page 11
- a 2-Field,1-Column page 12
- a Treasure Map page 10



Also shown are the green protractor and rulers on Side 1 that are enabled or disabled through the General Settings menu. Each of these pages will be described in detail as we proceed. Printing and folding instructions are described in an earlier section.

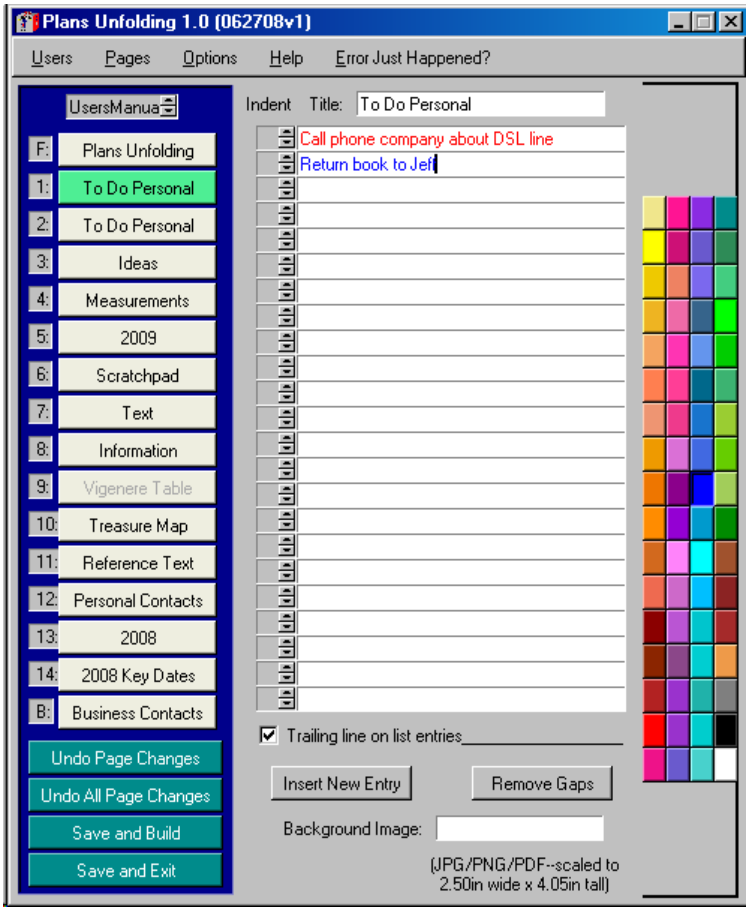
As mentioned, you can change these page types through a menu option, subject to the following restrictions :

- Calendars are only allowed on pages 5 and 13
- The Vigenere Cipher Table is only allowed on page 9, and the Cipher Text is only allowed on page 8
- The Treasure Map is only allowed on page 10
- The Astronomy page (not in the default layout) is only allowed on page 11

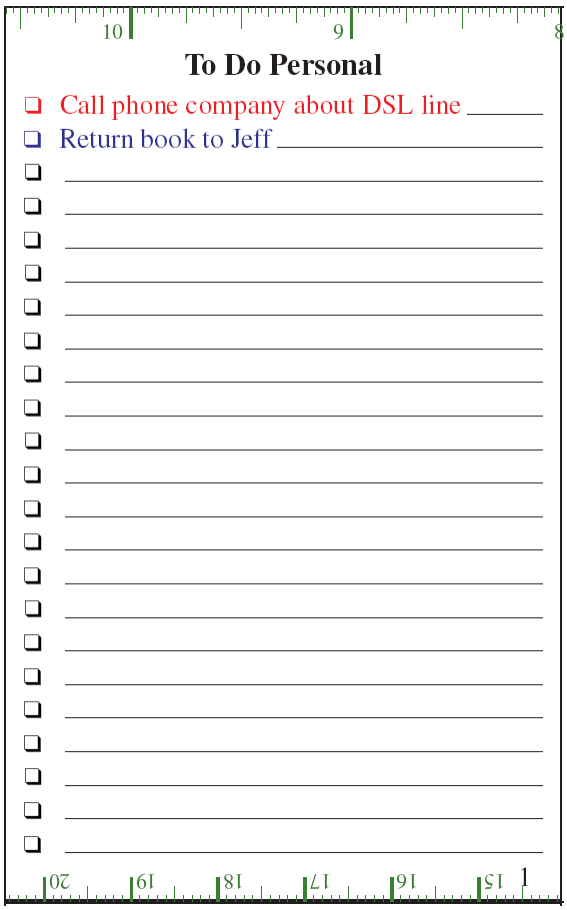
The page types directly supported by the interface are described in the sections below.

List Page Type

The List page type is the one you first encounter when you launch *Plans Unfolding* after creating a new user (on page 1). There are 23 fields for entering list items in addition to a title field. Clicking on a color swatch while the cursor is in any one of these fields sets the text to that color. A spinbox to the left of each entry can be rotated to make that entry indented—you have to remember throughout that LaTeX disregards extra spaces you insert unless you use LaTeX spacing commands, and LaTeX commands can only be used in Text pages.



There is a checkbox below the entries to produce trailing lines at the end of the text in any entry, which is the default setting. The **Insert New Entry** button create an empty entry at the entry containing the cursor, shifting the later entries down one entry if possible. This allows you to insert an entry anywhere you want. The **Remove Gaps** button will move up all entries so all empty entries lie at the bottom—this is also automatically done when the page is exited. Insert a space in a entry to reserve an empty line in the printout.

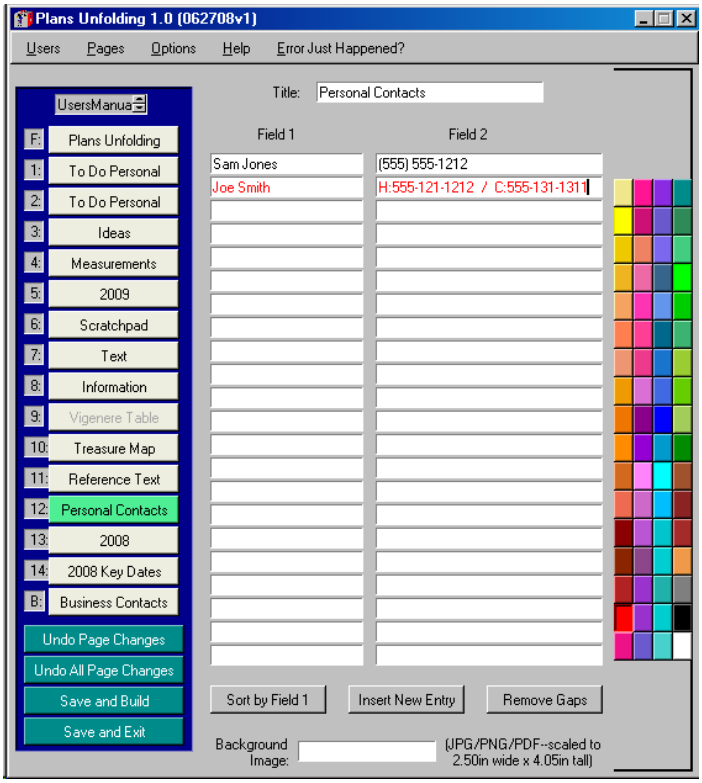


The actual printed page is shown on the right. The ruler pieces along the top and bottom are from the Ruler option in the General Settings menu., which produces rulers that cross this page 1.

The boxes to the left of the entries can be used to check off items on your printed organizer as they are completed. All entries are remembered by the *Plans Unfolding* application between sessions, but remember that you can pencil in new entries as desired between sessions as you carry the organizer with you.

2-Field,1-Column Page Type

The 2-Field,1-Column page type produces what is commonly seen as a phone list, say for a Business Contacts or Personal Contacts page. On each line the first field is printed on the left end, then a dotted underscore connects it to the second field on the right end. Beyond this, it is generally useful for matching sets consisting of two items.



There are 22 rows for entering items in addition to a title field. Clicking on a color swatch while the cursor is in any one of these fields sets the text to that color (the entire row for the Field entries). The **Sort by Field 1** button will sort all the rows alphabetically by Field 1, which useful it this is a phone list and you add contacts at the bottom. The **Insert New Entry** button create an empty entry at the entry containing the cursor, shifting the later entries down one entry if possible. This allows you to insert an entry anywhere you want. The **Remove Gaps** button will move up all entries so all empty entries lie at the bottom—this is also automatically done when the page is exited.

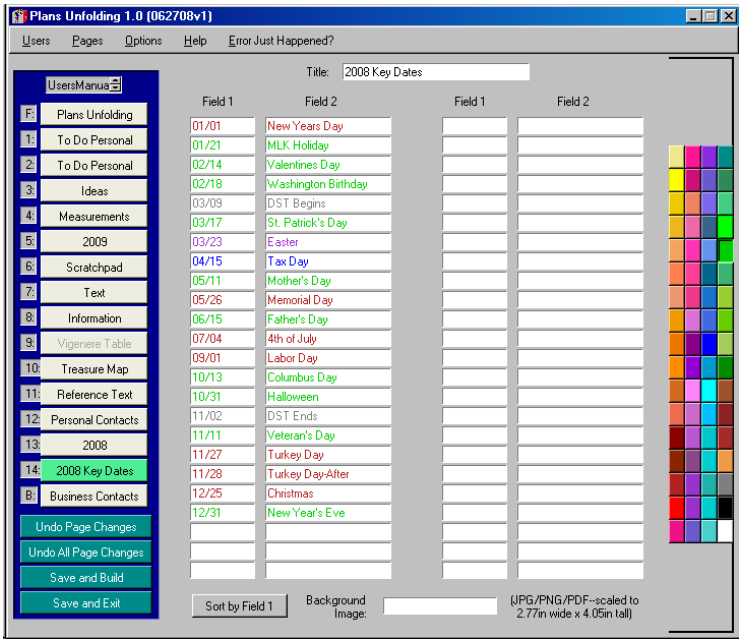
Business Contacts

Watson, Steve . . . w222-494-7538 / h444-289-8453
Williams, George w333-777-7777

B

2-Field,2-Column Page Type

The 2-Field,2-Column page type produces two columns, balanced in height to within one line, of two fields separated by a space. It is generally useful for matching a large number of sets consisting of two related items that are relatively short in length.



There are 48 rows for entering items in addition to a title field. Clicking on a color swatch while the cursor is in any one of these fields sets the text to that color (the entire row for the Field entries). The **Sort by Field 1** button will sort all the rows alphabetically by Field 1. Any gaps are automatically removed when the page is exited.

Keep in mind that if your fields are too long the line will wrap around in the output, limiting the number of rows you can use without running off the bottom of the page.

The best thing about this page is that you can mark dates in any Calendar page and export them to any 2-Field,2-Column page, which is why the page shown here for a new user consists of a Key Dates listing. You can either replace all fields with the Date/Comment pairs or add them to the end of existing pairs—see the Calendar page description for the procedure.

This page type is generally useful for a compact listing of paired data.

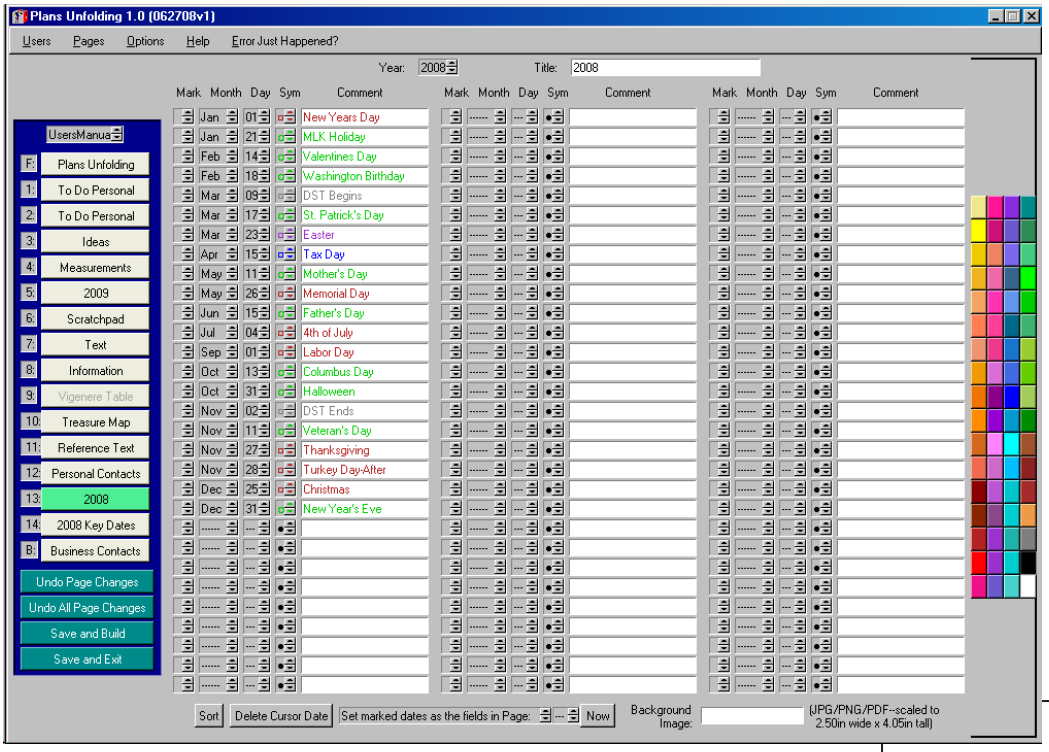
2008 Key Dates

01/01 New Years Day	07/04 4th of July
01/21 MLK Holiday	09/01 Labor Day
02/14 Valentines Day	10/13 Columbus Day
02/18 Washington Birthday	10/31 Halloween
03/09 DST Begins	11/02 DST Ends
03/17 St. Patrick's Day	11/11 Veteran's Day
03/23 Easter	11/27 Turkey Day
04/15 Tax Day	11/28 Turkey Day-After
05/11 Mother's Day	12/25 Christmas
05/26 Memorial Day	12/31 New Year's Eve
06/15 Father's Day	

14

Calendar Page Type

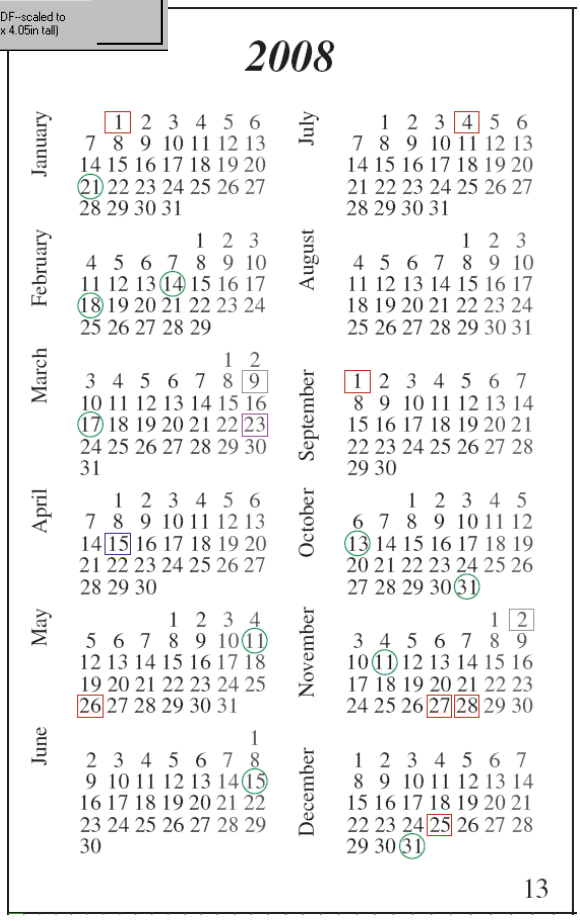
You can have up to 2 Calendar pages, and they must be located on pages 5 and 13, although these pages can be reassigned to other page types. Separately maintained calendars are available from 2008 through 2020, and these calendars are selected with the **Year** spinbox at the top of the interface. You can even have the same year for both pages 5 and 13 if you wish.



Use the **Month**, **Day** and **Sym** spinboxes on any of the 90 rows to create date "tags" on the output calendar. These tags can be hollow circles, filled circles, hollow squares or filled squares. You can enter a Comment for the date next to these. Clicking on a color swatch while the cursor is in any Comment field sets the tags and comment to that color.

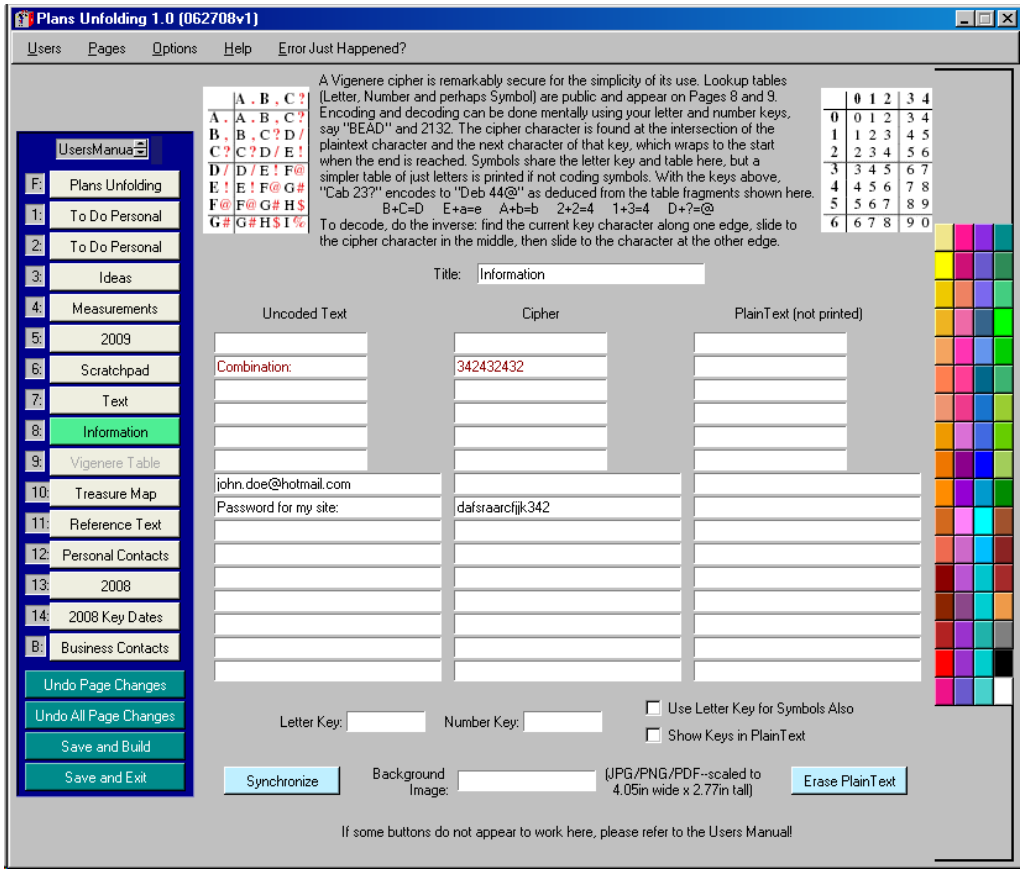
The **Sort** button will sort the dates, but this also occurs when you exit the page. The **Delete Cursor Date** button removes the tag entries for the date containing the cursor in the Comment field.

You can also replace the fields in any 2-Field,2-Column page with selected dates and their comments, or add these fields to the end of existing ones in such a page. First you mark the dates to export using the **Mark** spinbox to the left of the dates (these mark settings are remembered). Then select whether to add or replace the fields using the text spinbox at the bottom, then use the next spinbox to select from among your 2-Field,2-Column pages that are listed there, and then click on the **Now** button. This produces output like that seen in the 2-Field,2-Column example in the previous section. By default, the Key Dates page is located right next to the 2008 Calendar when the organizer is folded, which is convenient since the same color scheme is maintained between the text listing and the actual calendar. I find it useful to first replace the fields in my Key Dates page with marked dates for the remainder of 2008, then enter a row in the Key Dates page that says "2009 Dates", then go to the 2009 Calendar and add dates for the first part of 2009 to the end of the Key Dates page.



Cipher Table and Cipher Text Page Types

The combination of the Cipher Table and Cipher Text page types provides a unique way of encoding and decoding private information very quickly, without writing anything down. This utilizes the Vigenere Cipher, a remarkably secure encryption method given the simplicity of its use. To my knowledge no other paper organizer has such a feature, one that I find extremely useful for carrying passwords and other protected information. The Cipher Table can only be implemented on page 9 and the Cipher Text on page 8, but you can reassign these pages to different page types if you want. Normally you would reassign both pages, as these two page types work together.


Plans Unfolding 1.0 (062708v1)
Users
Pages
Options
Help
Error Just Happened?
A Vigenere cipher is remarkably secure for the simplicity of its use. Lookup tables (Letter, Number and perhaps Symbol) are public and appear on Pages 8 and 9. Encoding and decoding can be done mentally using your letter and number keys, say "BEAD" and 2132. The cipher character is found at the intersection of the plaintext character and the next character of that key, which wraps to the start when the end is reached. Symbols share the letter key and table here, but a simpler table of just letters is printed if not coding symbols. With the keys above, "Cab 23?" encodes to "Deb 44@" as deduced from the table fragments shown here.
B+C=D E+a=e A+b=b 2+2=4 1+3=4 D+?=@
To decode, do the inverse: find the current key character along one edge, slide to the cipher character in the middle, then slide to the character at the other edge.
UsersManua
F: Plans Unfolding
1: To Do Personal
2: To Do Personal
3: Ideas
4: Measurements
5: 2009
6: Scratchpad
7: Text
8: Information
9: Vigenere Table
10: Treasure Map
11: Reference Text
12: Personal Contacts
13: 2008
14: 2008 Key Dates
B: Business Contacts
Undo Page Changes
Undo All Page Changes
Save and Build
Save and Exit
Title: Information
Uncoded Text
Cipher
PlainText (not printed)
Combination:
342432432
john.doe@hotmail.com
Password for my site:
dafsraarcfjjk342
Letter Key:
Number Key:
Use Letter Key for Symbols Also
Show Keys in PlainText
Synchronize
Background Image:
(JPG/PNG/PDF--scaled to 4.05in wide x 2.77in tall)
Erase PlainText
If some buttons do not appear to work here, please refer to the Users Manual!

The Vigenere Cipher uses a public table and a privately-known key, so all you need to remember is a keyword of several characters, say 4-8 characters long, and a keynumber for encoding numbers. Of course you wouldn't use a common word or name—perhaps you would use a variation on a word in the dictionary or a Latin word or such. The keynumber shouldn't be, say, your house number, phone number or the first digits of pi, for example. Once you settle on a keyword and keynumber you can encode and decode text from within the interface or on-the-fly with your paper organizer.

The window describes the use of the Vigenere Cipher along the top, which we will discuss here. There are three columns of fields. The **Uncoded Text** column contains text that will appear unencrypted in a row of output—for example, this might be a label to indicate what the encrypted text refers to. The **Cipher** column contains the encrypted text that will appear next to the Uncoded Text in the row of output. The **PlainText** column contains the decrypted version of the Cipher text and is not printed in the organizer. There are 6 rows that can contain short text (because the number table lies next to them in the output) followed by 9 rows of longer text.

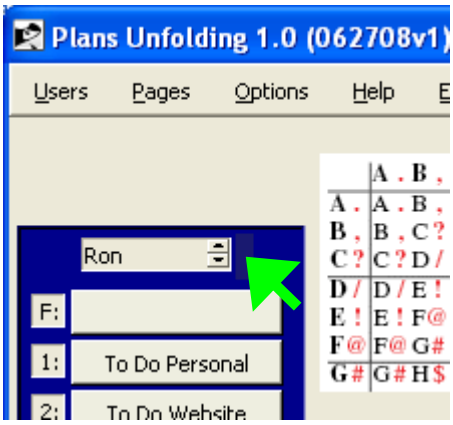
Cipher Text can be decoded within the interface, in which case the PlainText field will be populated with the decoded text. PlainText can also be encoded within the interface, in which case the Cipher Text field will be populated with the encoded text and will be printed in the organizer. If both fields are populated when the synchronization process is performed, the decryption of the Cipher Text will overwrite the PlainText field.

To perform the encryption/decryption process, first enter your keyword and (if you have numbers in your code) your keynumber in the **Letter Key** and **Number Key** fields, which will display only asterisks for the characters. The **Show Keys in PlainText** checkbox will display the actual characters in your keyword and keynumber, and the **Synchronize** button will synchronize the Cipher Text and PlainText fields in all the rows. If a keyword or keynumber has not been entered when the synchronization is initiated, question marks (?) will be output for those characters. Uppercase and lowercase are preserved in the output.

For security reasons in case you leave your computer unattended while your keyword/keynumber is entered, there is a two-step process to use the **Show Keys in PlainText** and **Synchronize** buttons!

> You **must** click on a small, almost imperceptibly darker rectangle just to the right of your user name in the upper spinbox after you check the **Show Keys in PlainText** button or click on the **Synchronize** button to make them work.

The **Synchronize** button will turn light red when clicked to indicate that it's "armed" waiting for this second step. The note at the bottom of the window is a clue to look in this User's Manual if the buttons don't seem to work.



The **Erase PlainText** button is a fast reset, clearing the PlainText column and setting the keyword and keynumber to display asterisks. When you exit and re-enter this page the PlainText is erased as well as your keyword and keynumber, which must be re-entered to synchronize.

There is also a checkbox to encrypt symbols (such as punctuation) as well as letters and numbers. I don't use this but some like it. This will be discussed in the following description of using the Vigenere Cipher.

Here are the organizer output pages for the Cipher Table and Cipher Text:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

6

	0	1	2	3	4	5	6	7	8	9
0	0	1	2	3	4	5	6	7	8	9
1	1	2	3	4	5	6	7	8	9	0
2	2	3	4	5	6	7	8	9	0	1
3	3	4	5	6	7	8	9	0	1	2
4	4	5	6	7	8	9	0	1	2	3
5	5	6	7	8	9	0	1	2	3	4
6	6	7	8	9	0	1	2	3	4	5
7	7	8	9	0	1	2	3	4	5	6
8	8	9	0	1	2	3	4	5	6	7
9	9	0	1	2	3	4	5	6	7	8

Information

8

Combination: 342432432

john.doe@hotmail.com

Password for my site: dafsraarcfjjk342

The Cipher Table page contains the public Vigenere lookup table that is used to map characters in your keyword with characters in the plaintext to get the coded characters, or vice-versa to decode characters. The lookup table for numbers is part of the Cipher Text page, which is why the first 6 rows of entry are so short in the interface. I use the title "Information" for this page because I also use it for unencoded information such as the email address in the example. (That's not the password for my site, by the way.)

It's easy to encode and decode using the Vigenere cipher, and it can be done mentally without writing down your key or any intermediate text! Each row starts a new encoding/decoding process with your keyword and keynumber.

Let's say your keyword is BEAD and your keynumber is 2192. The cipher character is found at the intersection of the plaintext character and the next character of that key, which wraps to the start when the end is reached. When **Use Letter Key For Symbols Also** is unchecked in the interface, all punctuation characters and any other ones that are not letters and numbers are simply output as themselves, and in this case the letter table is the simple one given above.

So say you have a password you want to write into your Cipher Text page of your organizer when you're not using the interface software. If the password is ***Turtle*** we locate the first letter of our keyword ***B*** along one edge of the lookup table, then find the letter that lies at the intersection of that row or column and the one corresponding to ***t***

along the other edge, and that's our cipher letter—looking at the table above we see that it will be ***U***. Let's denote this as:

B + T = U

Then we do the same for the next letter of the keyword ***E*** and of the plaintext ***u*** and we get ***y***, and so on. When we finish with the last letter **D** of the keyword we wrap around and continue with the first letter **B**.

E + u = y A + r = r D + t = w B + l = m E + e = i

In this way we enter ***Uyrwmi*** in the organizer for ***Turtle***. To decode ***Uyrwmi*** we do the reverse. We locate the first letter of our keyword ***B*** along one edge, slide along the row or column until we arrive at ***U***, and we see what character lies along the other edge, ***T*** in this case. It can all be done without writing down the key or text.

B – U = T etc.

Numbers use the keynumber (here 2192) and the number lookup table, so if the password were ***Tu2rtle84*** the digit **2** would be matched against the first digit of our keynumber **2** to produce **4**, the **8** would be matched with the second digit **1** to produce **9**, and the **4** would be matched with **9** to get **2**. In fact, when encoding the number lookup table simply produces the units digit of the sum of the key and plaintext digits, so you don't actually need the table. (You can also use this fact in reverse to easily decode a number.)

2 + 2 = 4 8 + 1 = 9 9 + 3 = 2

So we end up with ***Uy4rwmi92***.

When **Use Letter Key For Symbols Also** is checked in the interface, the more complicated Vigenere lookup table shown here is generated on page 9. It includes symbols along with the letters. Symbols act like letters in that they share the keyword with the letters, but matching is done between keyword letters and the symbols. So looking the letter-symbol table here, the phrase ***joe@this.com*** would produce

B + j = k
E + o = s
A + e = e
D + @ = %
etc.

to arrive at ***kse%uliv,gop*** as the ciphertext. Decoding is done in the same manner.

	A . B , C ?	D / E ! F @	G # H $ I %	J ^ K & L *	M (N) O -	P + Q = R _	S ; T : U '	V " W < X >	Y \ Z ~
A .	A . B , C ?	D / E ! F @	G # H $ I %	J ^ K & L *	M (N) O -	P + Q = R _	S ; T : U '	V " W < X >	Y \ Z ~
B ,	B , C ? D /	E ! F @ G #	H $ I % J ^	K & L * M (	N) O - P +	Q = R _ S ;	T : U ' V "	W < X > Y \	Z ~ A .
C ?	C ? D / E !	F @ G # H $	I % J ^ K &	L * M (N)	O - P + Q =	R _ S ; T :	U ' V " W <	X > Y \ Z ~	A . B ,
D /	D / E ! F @	G # H $ I %	J ^ K & L *	M (N) O -	P + Q = R _	S ; T : U '	V " W < X >	Y \ Z ~ A .	B , C ?
E !	E ! F @ G #	H $ I % J ^	K & L * M (	N) O - P +	Q = R _ S ;	T : U ' V "	W < X > Y \	Z ~ A . B ,	C ? D /
F @	F @ G # H $	I % J ^ K &	L * M (N)	O - P + Q =	R _ S ; T :	U ' V " W <	X > Y \ Z ~	A . B , C ?	D / E !
G #	G # H $ I %	J ^ K & L *	M (N) O -	P + Q = R _	S ; T : U '	V " W < X >	Y \ Z ~ A .	B , C ? D /	E ! F @
H $	H $ I % J ^	K & L * M (	N) O - P +	Q = R _ S ;	T : U ' V "	W < X > Y \	Z ~ A . B ,	C ? D / E !	F @ G #
I %	I % J ^ K &	L * M (N)	O - P + Q =	R _ S ; T :	U ' V " W <	X > Y \ Z ~	A . B , C ?	D / E ! F @	G # H $
J ^	J ^ K & L *	M (N) O -	P + Q = R _	S ; T : U '	V " W < X >	Y \ Z ~ A .	B , C ? D /	E ! F @ G #	H $ I %
K &	K & L * M (	N) O - P +	Q = R _ S ;	T : U ' V "	W < X > Y \	Z ~ A . B ,	C ? D / E !	F @ G # H $	I % J ^
L *	L * M (N)	O - P + Q =	R _ S ; T :	U ' V " W <	X > Y \ Z ~	A . B , C ?	D / E ! F @	G # H $ I %	J ^ K &
M (	M (N) O -	P + Q = R _	S ; T : U '	V " W < X >	Y \ Z ~ A .	B , C ? D /	E ! F @ G #	H $ I % J ^	K & L *
N)	N) O - P +	Q = R _ S ;	T : U ' V "	W < X > Y \	Z ~ A . B ,	C ? D / E !	F @ G # H $	I % J ^ K &	L * M (
O -	O - P + Q =	R _ S ; T :	U ' V " W <	X > Y \ Z ~	A . B , C ?	D / E ! F @	G # H $ I %	J ^ K & L *	M (N)
P +	P + Q = R _	S ; T : U '	V " W < X >	Y \ Z ~ A .	B , C ? D /	E ! F @ G #	H $ I % J ^	K & L * M (	N) O -
Q =	Q = R _ S ;	T : U ' V "	W < X > Y \	Z ~ A . B ,	C ? D / E !	F @ G # H $	I % J ^ K &	L * M (N)	O - P +
R _	R _ S ; T :	U ' V " W <	X > Y \ Z ~	A . B , C ?	D / E ! F @	G # H $ I %	J ^ K & L *	M (N) O -	P + Q =
S ;	S ; T : U '	V " W < X >	Y \ Z ~ A .	B , C ? D /	E ! F @ G #	H $ I % J ^	K & L * M (	N) O - P +	Q = R _
T :	T : U ' V "	W < X > Y \	Z ~ A . B ,	C ? D / E !	F @ G # H $	I % J ^ K &	L * M (N)	O - P + Q =	R _ S ;
U '	U ' V " W <	X > Y \ Z ~	A . B , C ?	D / E ! F @	G # H $ I %	J ^ K & L *	M (N) O -	P + Q = R _	S ; T :
V "	V " W < X >	Y \ Z ~ A .	B , C ? D /	E ! F @ G #	H $ I % J ^	K & L * M (	N) O - P +	Q = R _ S ;	T : U '
W <	W < X > Y \	Z ~ A . B ,	C ? D / E !	F @ G # H $	I % J ^ K &	L * M (N)	O - P + Q =	R _ S ; T :	U ' V "
X >	X > Y \ Z ~	A . B , C ?	D / E ! F @	G # H $ I %	J ^ K & L *	M (N) O -	P + Q = R _	S ; T : U '	V " W <
Y \	Y \ Z ~ A .	B , C ? D /	E ! F @ G #	H $ I % J ^	K & L * M (	N) O - P +	Q = R _ S ;	T : U ' V "	W < X >
Z ~	Z ~ A . B ,	C ? D / E !	F @ G # H $	I % J ^ K &	L * M (N)	O - P + Q =	R _ S ; T :	U ' V " W <	X > Y \

These are all done automatically when you synchronize using the software interface, of course, but this Vigenere Cipher allows you to decode and code as you like away from the computer! And it can be done without writing anything down at all.

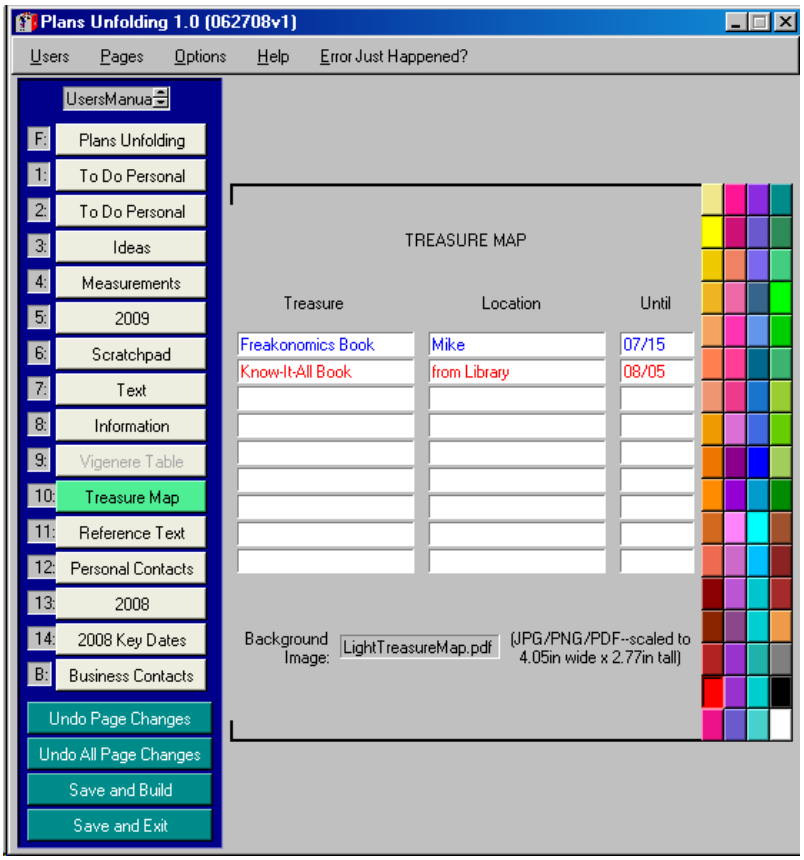
Choosing and Using Keys

I would like to provide some important tips on using this cipher, or for any other cipher for that matter. First, as I mentioned before you don't want to pick a dictionary word for your keyword—use a variation of your own. Also,

don't choose the initial digits, say, of a constant as your keynumber. These are susceptible to cracking. And look at the ***kse%uliv,gop*** ciphertext we just generated—this certainly looks like an email address with the locations of the punctuation, so the lookup table can be used to identify the keyword letters that were matched with the @ and . characters when this was encoded, and *com* is a good guess for the last three letters—in this case add some characters (such as x's here and there) or otherwise disguise this character pattern. Don't directly encode phone numbers or Social Security numbers without rotating or otherwise distorting the digit positions—someone could guess the phone prefix, or if they know where you were born they would probably know the first 3 digits of your Social Security number. Driver's license numbers are equally bad—in some states such as Illinois the digits are assigned based on the phonetic spelling of your last name, date of birth, gender, etc., so even if your driver's license card is secure and even if no one sees your number on your checkbook or other public places, the number can be deduced from state algorithms. Once your keyword or keynumber is cracked, it is cracked for all the entries on your organizer page. Avoid using an unencoded title that provides clues to what the ciphertext is about. And one more thing: I wouldn't recommend a keyword that starts with the letter A, as this encodes any letter to itself and might provide a clue on what the item might be—it's fine to have an A in the middle of your keyword, though. Use a keyword/keynumber of at least 5 characters, and don't pick one with more than one repeated letter. For additional security you might consider either eliminating spaces between words or randomly adding them within words to disguise word patterns.

Treasure Map Page Type

The Treasure Map page is a place to keep notes on where you left something, who you lent something to, when a book needs to be returned, and so forth—any item you want to remember to do something with. The interface is simple, consisting of 9 rows of three columns (**Treasure**, **Location** and **Until**) whose contents will overlay the Treasure Map graphic in the proper positions in the PDF output.

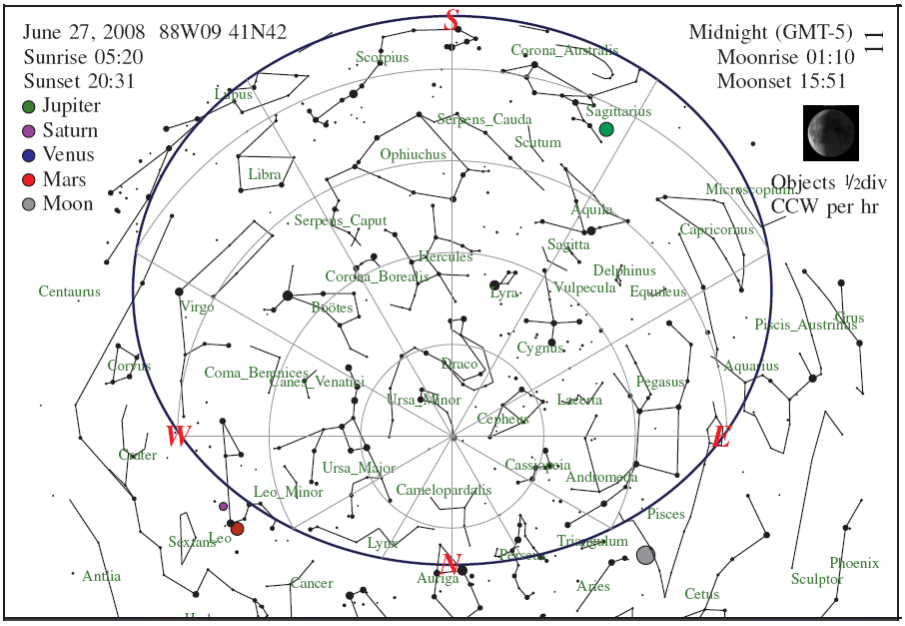


Again, you can change the font color of any row by placing the cursor in any field and depressing a color swatch. The Background Image filename is not editable since it is specifically associated with the text positions, and this is the one background image that does not need to be in the BackgroundImages directory (it is installed in the BuildArea directory).

The Treasure Map page can only be selected for page 10, but you can use this page for a different type of page if you prefer.

Astronomy Page Type

The Astronomy page is not selected for the default new user despite the amount of work that went into coding it, since most users may not want it. It can be selected (for page 11 only) through the Pages→Change Page Types menu option. It will calculate and display a planispheric projection of the sky at midnight local time on the date that the organizer is built—the objects will rotate counter-clockwise ½-division per hour after midnight (or ½-division clockwise per hour before midnight). It is designed for any latitude in the Northern Hemisphere. The printout shows on a grid marked at 30-deg intervals the locations in the sky of:



- the brightest stars
- major constellations and constellation names
- the planets Jupiter, Saturn, Venus and Mars
- the Moon

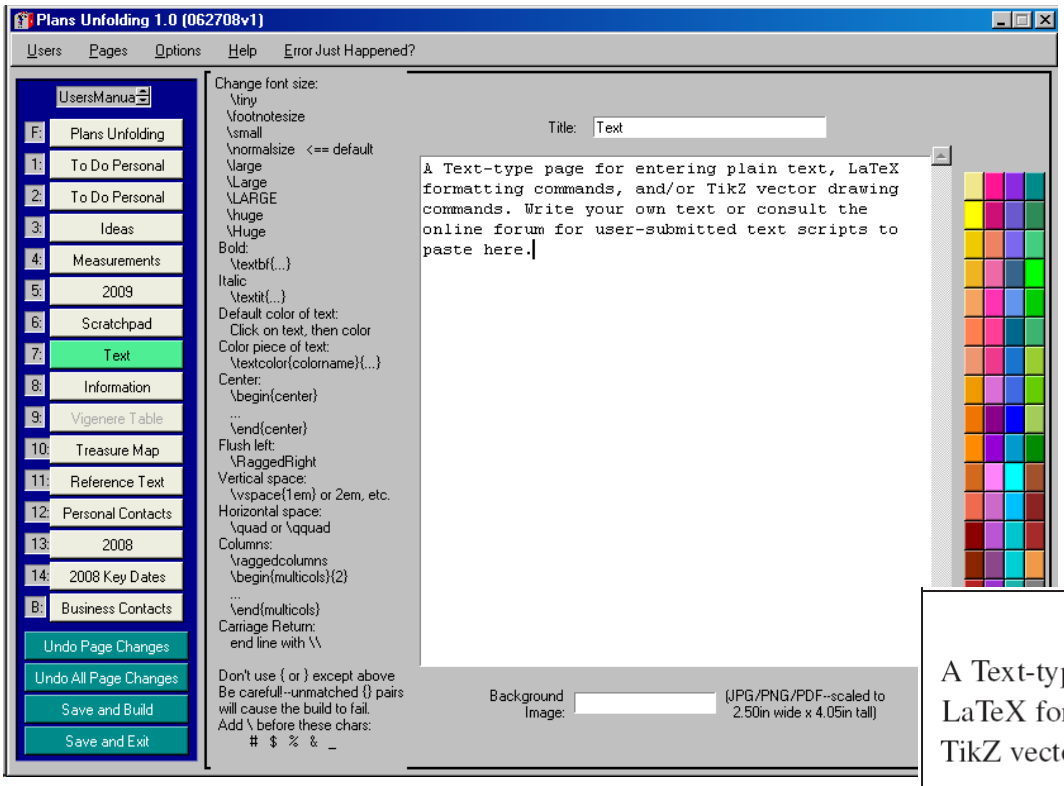
It also includes the times of sunrise/sunset and moonrise/moonset as well as the phase of the moon. Note that the moonset time will be the next one *after* the moonrise, so that time might be early on the next day—this is different from almanacs which will generally provide the moonset earlier than the moonrise time if the previous moon passage ended before moonrise on that day.

This sky map works like a planisphere you would buy in a store. You hold it up over your head with the center of the polar grid pointing at the North Pole, and the radial grid lines emanate from it. The horizon for a 42-deg latitude is shown—this will change a little at other latitudes so consider the horizon to be a rough guide. This map is quite accurate but you should never trust this map for anything important—it is only for use as a casual sky map for locating stars and planets.

This page is not editable in the interface. You do need to set your latitude, longitude and timezone via the Options→General Settings menu. The coordinates are entered in decimal degrees, not degrees and minutes, although the page prints out this information in degrees and minutes. Longitude is factored into the calculations to produce times that will match clock time at your location. You have to set the timezone difference from GMT in that same menu, and you also have to change this to account for any Daylight Savings Time (so Central Standard Time of GMT-6 needs to be changed to GMT-5 during the summer months).

Text Page Type

Text pages have a **much** more interesting role in this application than you might think at first. The simplest use of this type is simply typing plain text into the scrollable entry field and typing a title in the Title field, as shown here for page 7. Clicking on a color swatch while the cursor is in either of these fields sets the text to that color. The text is automatically wrapped in the page (and will breezily continue off the bottom of the page if the text is too long).



This is rather boring-looking, but it can be useful. Remember again that LaTeX ignores extra spaces and even carriage returns unless you leave a blank line between two lines. This can be overcome by using LaTeX commands. In fact,

Text Pages are the only pages that allow LaTeX commands!

and this is exactly what makes Text pages so powerful. Not only can you format your text, tables, math equations, and so forth if you know LaTeX, but you can also insert vector drawings using the TikZ drawing package! And best of all,

Custom Text Pages can be shared by simply pasting in the text!

In fact, there is an online Custom Page Script Collection and a Custom Page Script Screenshot gallery to share these creations—please see the *Support* section for information about this. So this means that *Plans Unfolding* is user-extensible and

Users can create custom page layouts and share them!

Of course you have to know something about LaTeX commands, and if you want higher-quality illustrations than produced by background images you will want to know something about the TikZ drawing package for LaTeX. So let's talk awhile about LaTeX commands and what they can do for you.

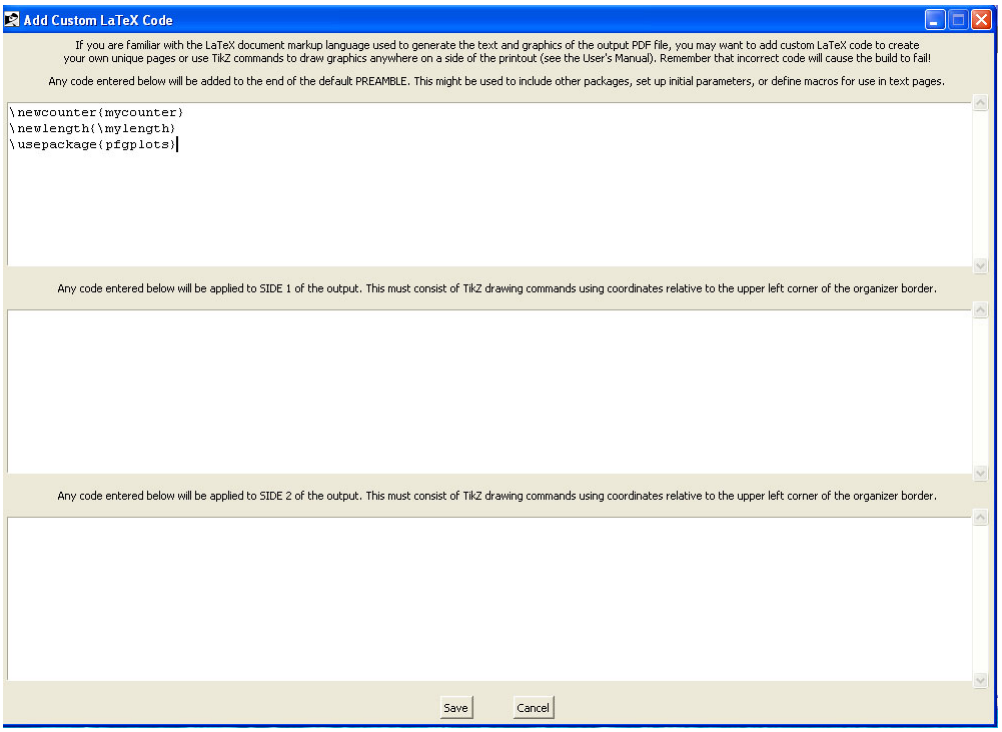
Text

A Text-type page for entering plain text, LaTeX formatting commands, and/or TikZ vector drawing commands. Write your own text or consult the online forum for user-submitted text scripts to paste here.

7

LaTeX is a typesetting language based on the TeX language created by Donald Knuth back in the 1970's. Math and science students often learn it because it was designed to typeset mathematical symbols and equations very elegantly for books and journal articles, and many journals require submissions in that format. But many, many add-on packages have been created by users to provide extra typesetting and drawing features. There are books on LaTeX and many online resources, but a really great introduction to writing LaTeX scripts can be found in the free online publication *The Not So Short Introduction to LaTeX 2e* found at http://www.ctan.org/tex-archive/info/lshort/english/lshort.pdf . Note that you can ignore Chapter 1 on document layout (*Plans Unfolding* is doing that behind the scenes)—you just want to use the typesetting commands. An excellent overview of math and special symbol typesetting is at ftp://ftp.ams.org/pub/tex/doc/amsmath/short-math-guide.pdf . And a very comprehensive list of special symbols available in a variety of packages can be found at http://www.ctan.org/tex-archive/info/symbols/comprehensive/symbols-letter.pdf .

As mentioned, extra LaTeX commands are available via add-on packages, and there are many of these. If you want to add a package that is not in the base *Plans Unfolding* set (one way to tell if it is missing is to try a command and see if you get a build error), then add the following line to the PREAMBLE section of the window displayed with the Options→Add Custom LaTeX Code menu function:


Add Custom LaTeX Code
If you are familiar with the LaTeX document markup language used to generate the text and graphics of the output PDF file, you may want to add custom LaTeX code to create your own unique pages or use TikZ commands to draw graphics anywhere on a side of the printout (see the User's Manual). Remember that incorrect code will cause the build to fail!
Any code entered below will be added to the end of the default PREAMBLE. This might be used to include other packages, set up initial parameters, or define macros for use in text pages.
\newcounter{mycounter}
\newlength{\mylength}
\usepackage{pfgplots}
Any code entered below will be applied to SIDE 1 of the output. This must consist of TikZ drawing commands using coordinates relative to the upper left corner of the organizer border.
Any code entered below will be applied to SIDE 2 of the output. This must consist of TikZ drawing commands using coordinates relative to the upper left corner of the organizer border.
Save
Cancel

\usepackage{*packagename*}

Many common packages are already included in this application, so their commands are available for you to use. Appendix A lists the packages included in the *Plans Unfolding* preamble along with links to their manuals, but remember that the base MiKTeX distribution contains many more.

So let's look at some examples of using LaTeX commands in a Text page. A short overview will allow you to take advantage of the most common typesetting features.

The left side of the Text page interface window displays a cheat sheet of common LaTeX commands for quick reference:

Change font size:
\tiny
\footnotesize
\small
\normalsize <== default
\large
\Large
\LARGE
\huge
\Huge

Bold:
\textbf{...}

Italic
\textit{...}

Default color of text:
Click on text, then color

Color piece of text:
\textcolor{colorname}{...}

Center:
\center{…}
or
\begin{center}
...
\end{center}

Flush left:
\RaggedRight

Vertical space:
\vspace{1em} or 2em, etc.

Horizontal space:
\quad or \qquad

Columns:
\raggedcolumns
\begin{multicols}{2}
...
\end{multicols}

Carriage Return:
end line with \\

Since the characters \ { } are used in commands, don't use these in the actual text you are formatting. The only exceptions are for these characters:

$ % &

which **must be preceded** by a \ character to distinguish them from LaTeX special characters.

Also, be careful—an unmatched set of {} will cause a build failure (The build output will probably provide the offending page number but it may produce a cryptic error message).

Finally, even though you can't access the standard editing menu by right-clicking the mouse (to cut, copy or paste text), you can swipe text and use the control keys to do these—this is very useful when copying and pasting scripts!

Ctrl-/ = select all text (equivalent to swiping the entire text)

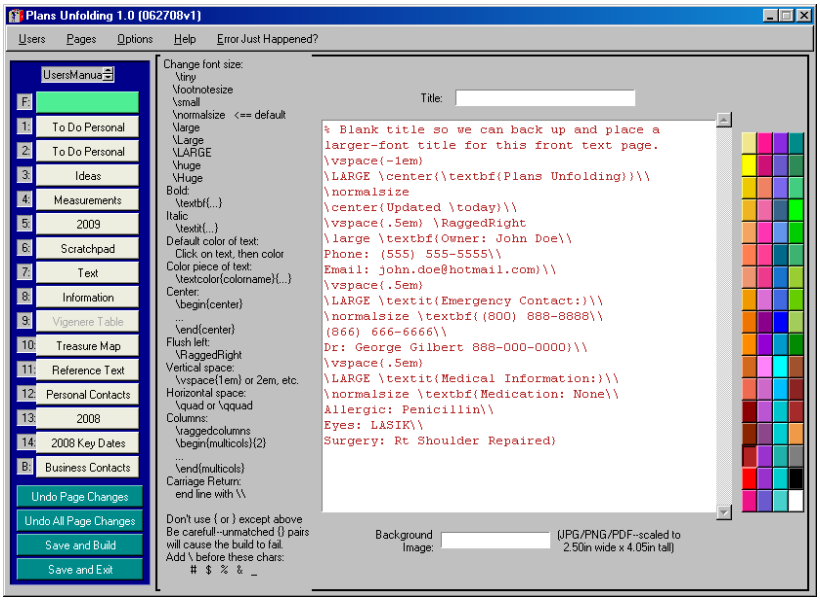
Ctrl-X = cut swiped text to Clipboard

Ctrl-C = copy swiped text to Clipboard

Ctrl-V = paste text from Clipboard

The one thing that will **NOT** work is Ctrl-Z as there is no Undo capability, unfortunately. However, you can Undo all changes to a page made during the current session of the application using the **Undo Page Changes** button along the left side of the interface window.

As an example, here's the default user page F (the front, or title page) of the organizer, showing some basic LaTeX formatting commands.




```
% Blank title so we can back up and place a
% larger-font title for this front text page.
\vspace{-1em}
\LARGE \center{\textbf{Plans Unfolding}}\\
\normalsize
\center{Updated \today}\\
\vspace{.5em} \RaggedRight
\large \textbf{Owner: John Doe\\
Phone: (555) 555-5555\\
Email: john.doe@hotmail.com}\\
\vspace{.5em}
\LARGE \textit{Emergency Contact:}\\
\normalsize \textbf{(800) 888-8888\\
(866) 666-6666\\
Dr: George Gilbert 888-000-0000}\\
\vspace{.5em}
\LARGE \textit{Medical Information:}\\
\normalsize \textbf{Medication: None\\
Allergic: Penicillin\\
Eyes: LASIK\\
Surgery: Rt Shoulder Repaired}
```

Text after a % on a line are just comments (the first line wraps around so it appears as two lines, but there is no carriage return).

We want a larger title for this front page than provided by the title entry field, so we leave that field blank and then back up with a negative vertical space. An *em* is a unit equal to the width of an *m* in current font, while a *pt* is a unit roughly equal to 1/3 of a mm (called a point). Here we back up 1em, set the font to LARGE, and center a bold **Plans Unfolding** title followed by a carriage return (\\).

Then we return the text to the default normal size and center the Update date using the \today command. We do a carriage return and a drop of .5em and set the text to left justification with the \RaggedRight command. The font size is set to large and we enter the personal info all in bold.

We repeat the drop, set the font to LARGE and print Emergency Contact in italic. You can see how the rest of the page follows the same process.

The default color for the text was set by clicking on the text in the window and then on a color swatch. However, we could have changed the color of any piece of text using the \textcolor command, as in \textcolor{Blue}{*text*} or for the title:

```
\center{\textbf{\textcolor{Blue}{Plans Unfolding}}}\\
```

We can also override the default text color by entering

```
\color{Blue}
```

Plans Unfolding

Updated June 27, 2008

Owner: John Doe
Phone: (555) 555-5555
Email: john.doe@hotmail.com

Emergency Contact:
(800) 888-8888
(866) 666-6666
Dr: George Gilbert 888-000-0000

Medical Information:
Medication: None
Allergic: Penicillin
Eyes: LASIK
Surgery: Rt Shoulder Repaired

F

Only valid LaTeX "dvips" colors are allowed—for convenience, these names appear along the top of the interface when the mouse hovers over the corresponding color swatches. These 68 valid colors are:

> GreenYellow, Yellow, Goldenrod, Dandelion, Apricot, Peach, Melon, YellowOrange, Orange, BurntOrange, Bittersweet, RedOrange, Mahogany, Maroon, BrickRed, Red, OrangeRed, RubineRed, WildStrawberry, Salmon, CarnationPink, Magenta, VioletRed, Rhodamine, Mulberry, RedViolet, Fuchsia, Lavender, Thistle, Orchid, DarkOrchid, Purple, Plum, Violet, RoyalPurple, BlueViolet, Periwinkle, CadetBlue, CornflowerBlue, MidnightBlue, NavyBlue, RoyalBlue, Blue, Cerulean, Cyan, ProcessBlue, SkyBlue, Turquoise, TealBlue, Aquamarine, BlueGreen, Emerald, JungleGreen, SeaGreen, Green, ForestGreen, PineGreen, LimeGreen, YellowGreen, SpringGreen, OliveGreen, RawSienna, Sepia, Brown,Tan,Gray

These are case-sensitive—uppercase letters in the names must be written in uppercase.

The *lettrine* package of LaTeX is included here to provide dropped initial capitals (see Appendix A for a link to its documentation). We can replace the "Plans Unfolding" title line with one that uses a lettrine command. Here the *lines* parameter specifies the height of the dropped cap, the *lhang* parameter specifies how far it should extend beyond the left edge of the text, and the *lraise* parameter specifies how much it should be moved vertically. The first set of curly braces contains the dropped cap and the second contains any text that is printed in small caps after the dropped cap. The Blue color swatch was also now depressed in the interface. The text and the output page are shown below. (LaTeX uses true small caps that preserve character line width of the original, unlike, say, Word.)

```
% Blank title so we can back up and place a
% larger-font title for this front text page.
\vspace{-1em}
\LARGE
\center{\lettrine[lines=2, lhang=0.0, lraise=0.1]{P}{lans Unfolding}}\\
\normalsize
\center{Updated \today}\\
\vspace{.5em} \RaggedRight
\large \textbf{Owner: John Doe\\
Phone: (555) 555-5555\\
Email: john.doe@hotmail.com}\\
\vspace{.5em}
\LARGE \textit{Emergency Contact:}\\
\normalsize \textbf{(800) 888-8888\\
(866) 666-6666\\
Dr: George Gilbert 888-000-0000}\\
\vspace{.5em}
\LARGE \textit{Medical Information:}\\
\normalsize \textbf{Medication: None\\
Allergic: Penicillin\\
Eyes: LASIK\\
Surgery: Rt Shoulder Repaired}
```

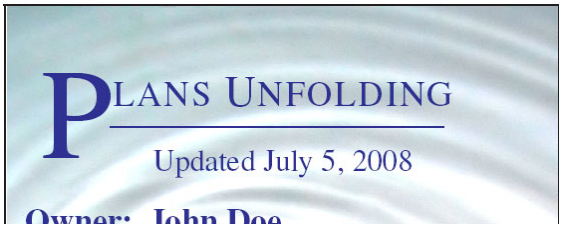
Perhaps it looks a little more elegant with a thin line (called a "rule") as the second line. In LaTeX the command for a rule is

```
\rule[raise-height]{width}{thickness}
```

where the optional length raise-height is the amount the rule is raised on the line and the mandatory lengths width and thickness specify these parameters. Adding the line

```
\rule[6.0pt]{3.85cm}{.5pt}
```

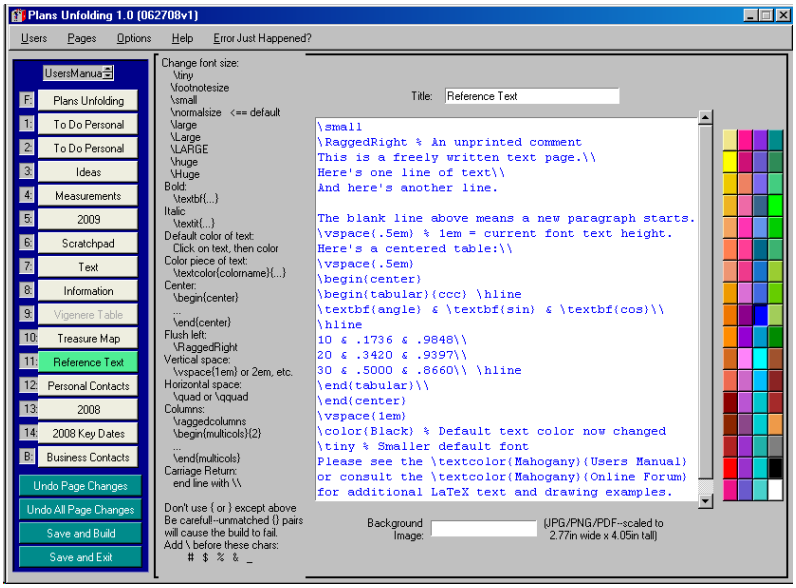
following the lettrine line changes the top of this page to the one shown on the lower right. (You don't have to have "Plans Unfolding" at all here.)



Here's an example where a table is created in a Text Page. Here we set the font size to small and left-justified and produce some unformatted text. The blank line means a new paragraph is started, and paragraph settings dictate what happens with a new paragraph. The *Plans Unfolding* preamble for LaTeX (found in the BuildArea folder) includes the following commands:

```
\parskip 1.5em
\parindent 0pt
```

So this new paragraph will start 1.5em down rather than a single line like those after \\ and it will have no indent. You can change these but please set them back to the above values at the end of the page.


Plans Unfolding 1.0 (062708v1)
Users Pages Options Help Error Just Happened?
Title: Reference Text
\small
\RaggedRight % An unprinted comment
This is a freely written text page.\\
Here's one line of text\\
And here's another line.
The blank line above means a new paragraph starts.
\vspace{.5em} % 1em = current font text height.
Here's a centered table:\\
\vspace{.5em}
\begin{center}
\begin{tabular}{ccc} \hline
\textbf{angle} & \textbf{sin} & \textbf{cos}\\
\hline
10 & .1736 & .9848\\
20 & .3420 & .9397\\
30 & .5000 & .8660\\ \hline
\end{tabular}\\
\end{center}
\vspace{1em}
\color{Black} % Default text color now changed
\tiny % Smaller default font
Please see the \textcolor{Mahogany}{Users Manual}
or consult the \textcolor{Mahogany}{Online Forum}
for additional LaTeX text and drawing examples.
Undo Page Changes
Undo All Page Changes
Save and Build
Save and Exit

Using the \begin{center}…\end{center} environment, a table is begun with three columns of centered text (ccc), where l or r would cause a column to be left or right justified. Then a simple horizontal line is drawn (\hline) and the first title row is entered in bold followed by another horizontal line. Each row is represented by entries separated by & symbols and ending in a \\ (a carriage return). A final horizontal line is printed and the table is ended. The rest should look familiar. So it's easy to fill in table values even though no specific interface exists for this page.

```
\small
\RaggedRight % An unprinted comment
This is a freely written text page.\\
Here's one line of text\\
And here's another line.

The blank line above means a new paragraph starts.
\vspace{.5em} % 1em = current font text height.
Here's a centered table:\\
\vspace{.5em}
\begin{center}
\begin{tabular}{ccc} \hline
\textbf{angle} & \textbf{sin} & \textbf{cos}\\
\hline
10 & .1736 & .9848\\
20 & .3420 & .9397\\
30 & .5000 & .8660\\
\hline
\end{tabular}\\
\end{center}
\vspace{1em}
\color{Black} % Default text color now changed
\tiny % Smaller default font
Please see the \textcolor{Mahogany}{Users Manual}
or consult the \textcolor{Mahogany}{Online Forum}
for additional LaTeX text and drawing examples.
```

Reference Text

This is a freely written text page.
Here's one line of text
And here's another line.
The blank line above means a new paragraph starts.
Here's a centered table:

angle	sin	cos
10	.1736	.9848
20	.3420	.9397
30	.5000	.8660

Please see the Users Manual or consult the Online Forum for additional LaTeX text and drawing examples.

11

We could have inserted vertical bars between columns by changing {ccc} to {|c|c|c|}, but in cases where data is pre-populated typographers look down on including the bars. We will insert vertical bars later when we create blank tables for penciling in data on the organizer. The *memoir* package provides more professional-looking tables by including \toprule, \middlerule and \bottomrule commands to produce varying (and customizable) line widths:

```
\begin{center}
\begin{tabular}{ccc} \toprule
\textbf{angle} & \textbf{sin} & \textbf{cos}\\
\midrule
10 & .1736 & .9848\\
20 & .3420 & .9397\\
30 & .5000 & .8660\\
\bottomrule
\end{tabular}\\
\end{center}
```

angle	sin	cos
10	.1736	.9848
20	.3420	.9397
30	.5000	.8660

With this introduction, let's take a detailed look at how we can create our own custom pages using LaTeX scripts in Text pages—text that can be shared via the online Custom Page Script Collectino and simply pasted into any Text page at all.

Customizing Your Organizer

The user-extensible nature of *Plans Unfolding* is one of its real strengths. You can use LaTeX commands or TikZ drawing commands, or both, to create any page design that you like in any Text page. Then, since these commands are all in plain ASCII text, you can share your design by uploading a text file containing the script to the Custom Page Script Collection online. A sample screenshot is also helpful, and this can be uploaded to the Custom Page Script Screenshot Gallery (linking them in each description field). Then others can download the script and paste it into any Text page of their own, and you can download scripts others have uploaded, editing them if they are not exactly what you want. **Feel free to skip this section if you are not interested in creating your own page designs.**

Vertical Tables

Let's start with a page containing a simple vertical table. (Letter output is assumed—A4 has slightly shorter/wider pages.) You may have been surprised that tables are not included in the standard interface page types. However,

- there are many possible table layouts;
- tables are usually empty so you can pencil in entries on the organizer as you carry it around, so entry widgets are not necessary;
- tables are very easy to create in Text pages; and
- it is extremely easy to add and edit data in a tables created in a Text page anyway

…so it makes more sense to provide examples and let you design the tables you want or download online creations.

Let's design a page containing a table for entering gasoline mileage and cost information. We enter "Vehicle Gas Mileage" in the Title field of any Text page—here we will assume a middle page in order to create a slightly wider table, so you would need to reduce the extra spacing a bit to fit it into an edge page. In fact, we'll use page 11, which is an "upside-down" page, to show that the script doesn't care what orientation the page is in (I will flip the resulting page right-side up here for convenience in reading it).

In the text below we first shift upward 0.3em toward the title to get more space for rows in the table. We start a centered environment with \begin{center}. Then we start a table of 5 columns separated by vertical lines, with any text printed in each column centered as denoted by the "c" in each column (it would be "l" for left-justified and "r"

for right-justified). We put a horizontal line at the top, then create the bolded entries on the first title row, separated by & symbols left-to-right as we saw earlier. Then another horizontal line and as many rows of empty text that we can fit without running off the bottom of the page, each terminated with a carriage return (\\) and a horizontal line. We add a line at the bottom, end the table, and then end the center environment. That's it.

```
\vspace{-.3em}
\begin{center}
\begin{tabular} {|c|c|c|c|c|}
\hline
\textbf{Date} & \textbf{Miles}& \textbf{Gal.} & \textbf{MPG} & \textbf{Cost}\\
\hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \
```

```
\hspace{.07cm}\textbf{MPG}\hspace{.07cm} &
\hspace{.07cm}\textbf{Cost}\hspace{.07cm}\\
\midrule
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\
\bottomrule
\end{tabular}\\
\end{center}
```

Let's color some columns. Adding "*>{\columncolor{colorname}}*" before the justification character (c, l or r) in the tabular argument will color that column. The *colorname* must be one of the 68 colors listed in this manual—these color names are also displayed above the color swatches in the interface when the mouse is hovered over the corresponding color. Note that all horizontal lines exist in the printout—the small magnification of the page displayed here doesn't show all the lines.

```
\vspace{-.3em}
\begin{center}
\begin{tabular}{
|>{\columncolor{Gray}}c|c|c|
>{\columncolor{SkyBlue}}c|
>{\columncolor{Goldenrod}}c|}
\toprule
\hspace{.07cm}\textbf{Date}\hspace{.07cm} &
\hspace{.07cm}\textbf{Miles}\hspace{.07cm} &
\hspace{.07cm}\textbf{Gal.}\hspace{.07cm} &
\hspace{.07cm}\textbf{MPG}\hspace{.07cm} &
\hspace{.07cm}\textbf{Cost}\hspace{.07cm}\\
\midrule
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\
\bottomrule
\end{tabular}\\
\end{center}
```

Vehicle Gas Mileage

Date	Miles	Gal.	MPG	Cost

11

Vehicle Gas Mileage

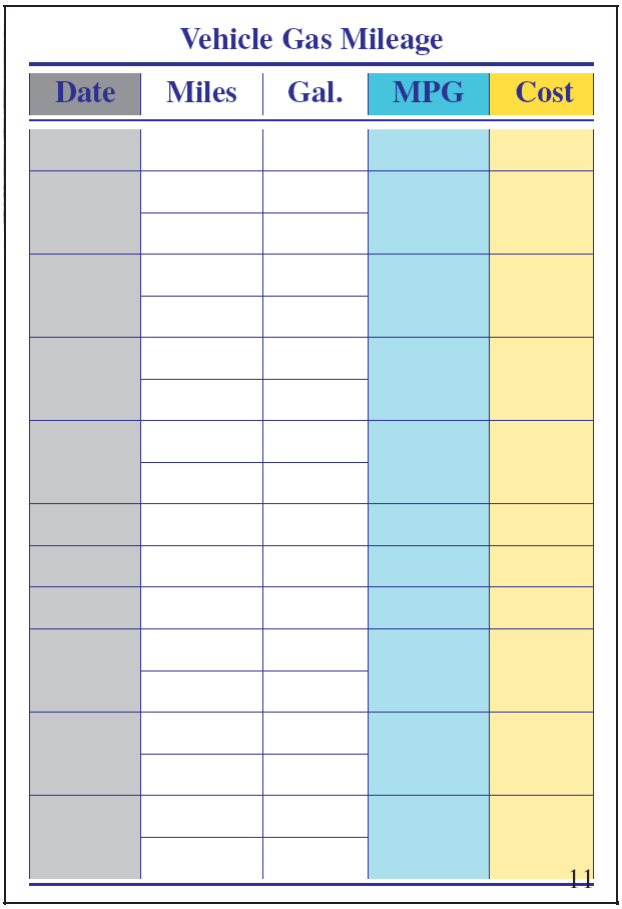
Date	Miles	Gal.	MPG	Cost

11

```
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\
\bottomrule
\end{tabular}\\
\end{center}
```

The colors in the data-entry cells look a little dark for penciling in entries on the paper organizer, so we might lighten the colors in those columns. You can lighten any of the standard 68 colors by appending the ! symbol followed by the percent of that color you want, so SkyBlue!50 is a much lighter version of SkyBlue. We will specifically leave the title colors dark with a \cellcolor{} command, which colors individual data cells.

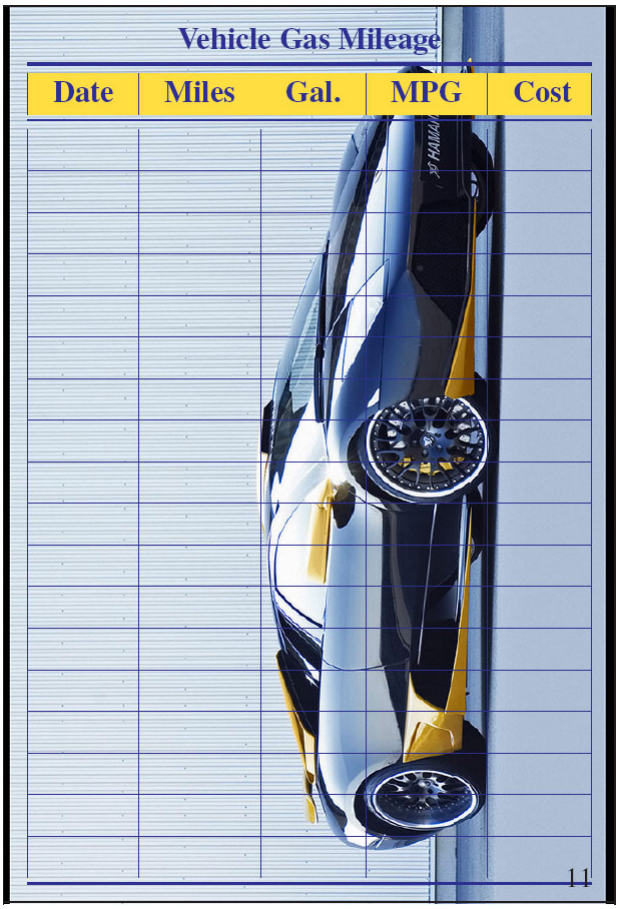
```
\vspace{-.3em}
\begin{center}
\begin{tabular}{
|>{\columncolor{Gray!50}}c|c|c|
>{\columncolor{SkyBlue!50}}c|
>{\columncolor{Goldenrod!50}}c|}
\toprule
\cellcolor{Gray}\hspace{.07cm}\textbf{Date}\hspace{.07cm} &
\hspace{.07cm}\textbf{Miles}\hspace{.07cm} &
\hspace{.07cm}\textbf{Gal.}\hspace{.07cm} &
\cellcolor{SkyBlue}\hspace{.07cm}\textbf{MPG}\hspace{.07cm} &
\cellcolor{Goldenrod}\hspace{.07cm}\textbf{Cost}\hspace{.07cm}\\
\midrule
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\
\bottomrule
\end{tabular}\\
\end{center}
```



This is nicer than the plain table. If we add a background image the colored columns will cover the image, though, so for our final version we eliminate these column colors, make all title cell colors Goldenrod using the \rowcolor{} command, and enter into the interface a background image of a car—here, use mine.

```
\vspace{-.3em}
\begin{center}
\begin{tabular}{|c|c|c|c|c|}
\toprule
\rowcolor{Goldenrod}\hspace{.07cm}\textbf{Date}\hspace{.07cm} &
\hspace{.07cm}\textbf{Miles}\hspace{.07cm} &
\hspace{.07cm}\textbf{Gal.}\hspace{.07cm} &
\hspace{.07cm}\textbf{MPG}\hspace{.07cm} &
```

```
\hspace{.07cm}\textbf{Cost}\hspace{.07cm}\\
\midrule
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\
\bottomrule
\end{tabular}\\
\end{center}
```



The background image should really be lightened up in a photo editor so pencil or pen marks are clearer on the paper, but it looks too cool to do that here. Again, the column widths will have to be reduced a bit for A4 or a corner page width (see later section for this). Also, ideally the image might be cropped a little narrower for a corner page so it isn't squeezed inward when it's included, although it actually still looks fine.

At first it seems that describing a typeset page using ASCII text is far inferior to a WYSIWIG word processing software program, which is often the initial reaction people have to LaTeX, but in fact the LaTeX engine is doing a lot of work for us. Centering is done automatically, lines are drawn automatically, columns are automatically widened if needed for any data we include, and so forth. Unless you are trying to customize the output in non-standard ways (and there are many more basic options and add-on packages of commands available), a LaTeX script is simple and surprisingly brief.

Horizontal Tables

You may have a table that takes up too much width to fit vertically on a page. You can create a horizontal table by exiting the default text node (which was created by the interface using a TikZ drawing command) and entering TikZ drawing commands directly into the page. TikZ is a very powerful package with an absolutely amazing users manual (see Appendix A for a link to this manual). This is just the briefest of introductions so you can start creating some custom pages right away.

To use TikZ drawing commands, you must first insert a line with the text BEGINDRAWING or BEGINDRAWING HORIZONTAL. You can have regular text and LaTeX commands prior to this line. This line is **required** before using **any** TikZ drawing commands, since the page text is contained within a TikZ draw node and that \draw command must be terminated before a new \draw command can be used. The line triggers the application to terminate the current text node and allows you the freedom to draw other text nodes, geometric shapes, plots and anything else that TikZ provides, anywhere on the page. NOTE: Horizontal drawings cannot be used on Page 7!

The default text node rotates the output either 0 degrees (for right-side-up pages) or 180 degrees (for upside-down pages), but here we need to embed the table in a text node that we rotate +/- 90 degrees, so we have to create our own text node to do this.

You have ultimate freedom in defining absolute sheet coordinates as provided in Appendix D, but if you want to make a node that will rotate horizontally on both a right-side-up and upside-down middle or corner page (in other words, a truly portable script for a horizontal layout) you must use the BEGINDRAWING HORIZONTAL, which the application will interpret depending on the orientation and location of the page. Not all TikZ draw constructs may work correctly under BEGINDRAWING HORIZONTAL, but certainly text nodes and line segments will. A text node that you want to end up horizontal on a page must be created as:

```
\draw (PX,PY) node[text width=\fullheight,inner sep=0pt,anchor=north west] {

    <Contents of node>

};
```

A TikZ node is a content block such as a textbox. Node contents are contained within squiggly brackets. All \draw commands are terminated by a semicolon. An "anchor" is point on the node that is placed on the given (x,y) coordinates. Possible *anchor* positions are the 8 points of a compass:

- north
- north east
- east
- south east
- south
- south west
- west
- north west

Drawing commands designed for vertical layouts can specify whatever anchor they desire (for example, north, south, east and west are convenient for centering a node regardless of the amount of text entered), but **for a script to display correctly in a horizontal layout using (PX,PY) relative coordinates, you must use "anchor=north west"** and use (PX,PY) coordinate offsets for where you want the upper-left corner of the node to lie.

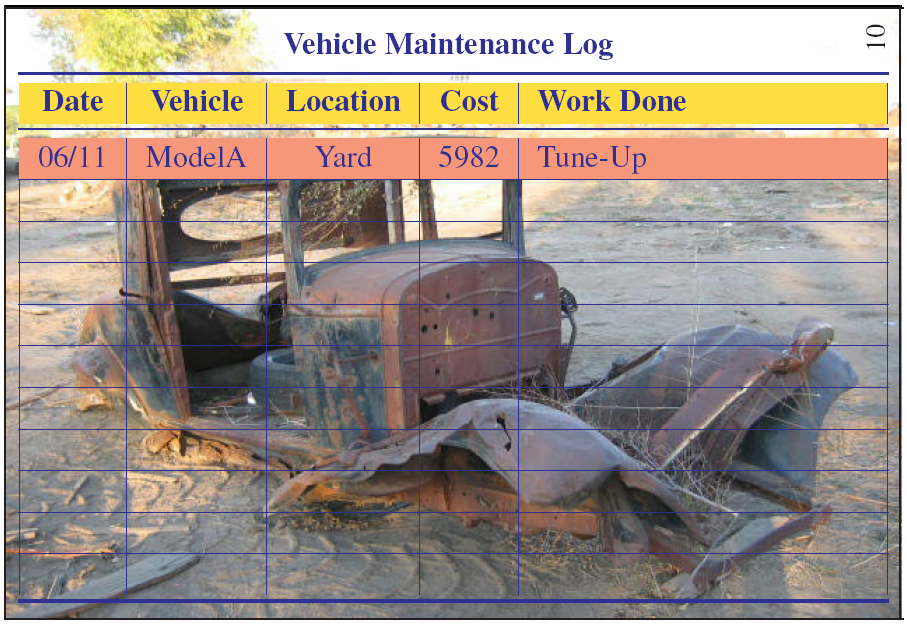
You can define the width of a text node (*text width*)and an inner separation of the text from that width (*inner sep*), as well as an angle to rotate the node (*rotate*) and other parameters we won't use here. Defining a text width in a text node is a very good idea, as carriage returns (\\) and lining up equations using the \begin{align} command require the text width to be defined. Here we just define it to be the total width available in a horizontal layout, or in other words **you should use *text width=\fullheight*** where the macro \fullheight is the height of the default vertical page.

Now PX and PY represent the horizontal x- and y-coordinates of the upper left corner of the current page (before flipping upside down for upside-down pages). The x-distance increases to the right and the y-value decreases as you proceed down the page. So (PX+150pt,PY-123pt) is the coordinates of a point 150pt from the left side of the page and 123pt down from the top. The use of PX and PY macros with offsets allow these custom page scripts to be pasted into any Text page.

So we have seen in the last example that you can create a single vertical table without entering the custom drawing mode, but to create a horizontal table you have to enter this mode. As an example of this, here's a page script for a horizontal table, also showing one colored row with entered data (you will have to remove one row to fit it in a corner page, and adjust the 2cm heading space for A4 output).

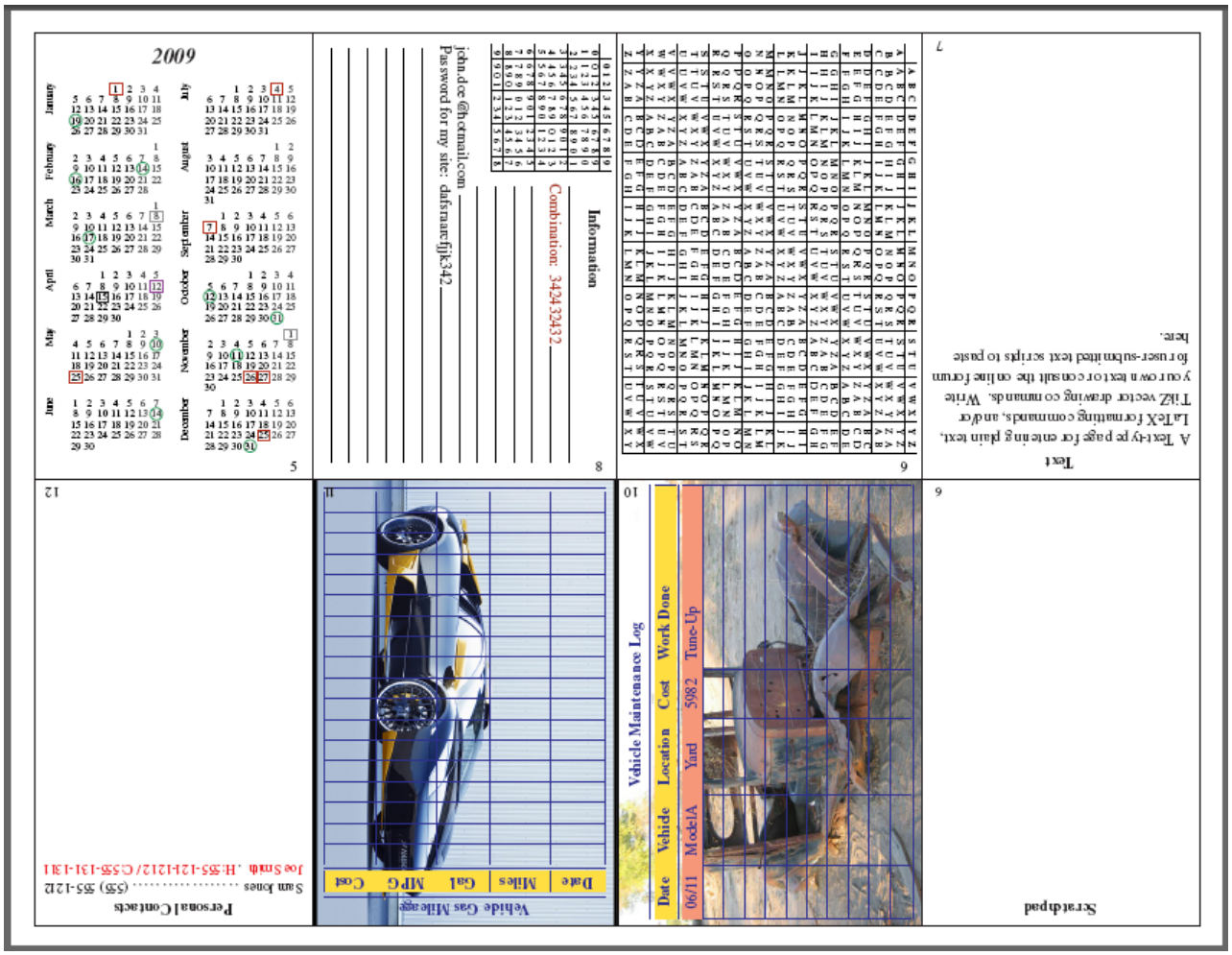
```
BEGINDRAWING HORIZONTAL
\draw (PX,PY) node[text width=\fullheight,inner sep=0pt,anchor=north west] {
\color{Blue}
\vspace{.8em}
\center{\textbf{Vehicle Maintenance Log}}
\vspace{.3em}
\begin{center}
\begin{tabular}{|c|c|c|c|l|}
```

```
\toprule
\rowcolor{Goldenrod}\textbf{Date} &
\textbf{Vehicle} &
\textbf{Location} &
\textbf{Cost} &
\textcolor{Blue}{\textbf{Work Done
\hspace{2cm}}}\\
\midrule
\rowcolor{Red!50} 06/11 & ModelA &
Yard & 5982 & Tune-Up\\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\
\bottomrule
\end{tabular}\\
\end{center}
};
```



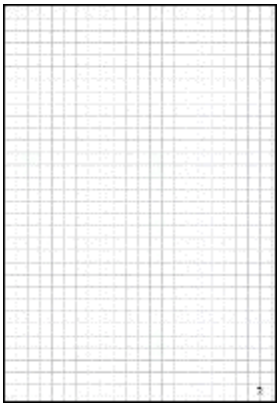
Oh, wait, this one's my car.

Here's what these two pages look like in context:



Writing Page Scripts that Work on Letter and A4, Middle and Corner Pages

We have seen in our table examples that the corner pages are narrower than the middle pages because they have an outside border. In the case of the Vehicle Gas Mileage table we produced a script that works on middle pages and mentioned that the column widths would have to be reduced to create the table for a corner page. This leads to either two separate page scripts for these two page locations or one script that asks the user to edit the information for the location of the particular page the script is being pasted into. This may not be a big deal to do, but there is a way to write a page script that handles both page locations without user intervention.



Starting with version 1.1 of Plans Unfolding, there is a mechanism that allows a single page script to change its variables or operation depending on internal knowledge of whether the page lies in a middle position or at a corner. Below is a page script from the online gallery to create a grid pattern on a page based on the *grid* drawing command in TikZ, a version that asks the user to change the coordinates of the grid extremes based on the page location.

```
BEGINDRAWING
% Use (PX,PY-293pt) grid (PX+200pt,PY) for middle pages.
% Use (PX,PY-293pt) grid (PX+182pt,PY) for corner pages.
\draw[xstep=0.125in,ystep=0.125in,line width=.5pt,Gray!25] (PX,PY-293pt) grid (PX+200pt,PY);
```

Starting with Version 1.1, a set of counters and lengths is populated with values for a particular page before the page is invoked, specifically for use in a portable page script:

Variable	Type	Preset Value
pagenum	counter	Current page number
ismiddlepage	counter	0 for an corner/edge page, 1 for a middle page
invertedpage	counter	0 for a right-side up page, 1 for an inverted page
fullwidth	length	Full width of the page
pagewidth	length	Display width of the page assuming a small (6pt) margin
indentedwidth	length	Display width of the page assuming a doubled (12pt) margin
pagemargin	length	Default margin (6pt)
printwidth	length	Default width for display on the page, set to pagewidth

You can set the page margin to a different value that you want in a page script using the \setlength{*name*}{*value*} command. Then the page script can call a new command **\calcprintwidth** that will re-calculate the available print width \printwidth as the \fullwidth of the page minus twice \pagemargin.

```
\setlength{\pagemargin}{8pt}        % sets left and right margin to 8pt
\calcprintwidth                     % Sets \printwidth based on \pagemargin and page location
```

By explicitly setting the value of \pagemargin even to the default of 6pt, the user can simply edit the value to whatever they want. The Music Staff Sheet example seen later offers this option to the user, and then uses the calculated \pagemargin value in order to draw the staff lines at the correct starting point and length across the page.

Using the set of page variables shown above, we can now rewrite the grid page script to work for either a middle or corner page without requiring any editing by the user. We use the \ifthenelse{*test*}{*command if test true*}{ *commands if test false*} command, where a counter value is read with \value{*countername*}:

```
BEGINDRAWING
% Works only under Plans Unfolding version 1.1 or higher.
\ifthenelse{\value{ismiddlepage} = 1}{
   \draw[xstep=0.125in,ystep=0.125in,line width=.5pt,Gray!25] (PX,PY-293pt) grid (PX+200pt,PY);
}{
   \draw[xstep=0.125in,ystep=0.125in,line width=.5pt,Gray!25] (PX,PY-293pt) grid (PX+182pt,PY);
}
```

Or better yet, use the \fullwidth variable listed in the table above, which is set for the current page based on whether it is a middle or edge page, and whether the output is on letter-size or A4 size paper.

```
BEGINDRAWING
\draw[xstep=0.125in,ystep=0.125in,line width=.5pt,Gray!25] (PX,PY-\fullheight) grid (PX+\fullwidth,PY);
```

For the Vehicle Gas Mileage table presented earlier, we can do a similar thing where we define a spacing length for the column headers based on the page width, and then use this variable in the table creation. We will discuss the creation of variables with \providelength{} and the use of variables in a walkthrough of the Music Staff Sheet script example a bit later. (Notice as before that we don't need a BEGINDRAWING line here when creating a table since we are using LaTeX commands without any TikZ \draw commands.) The header spacing \myspace assigned below is correct for letter-size paper.

```
% Vehicle Gas Mileage Table
% Works only under Plans Unfolding version 1.1 or higher.
%
\providelength{\myspace}
\ifthenelse{\value{ismiddlepage} = 1}{
   \setlength{\myspace}{0.07cm}  % Middle page
}{
   \setlength{\myspace}{0.01cm}  % Corner page
}
\vspace{-.3em}
\begin{center}
\begin{tabular}{|c|c|c|c|c|}
\toprule
\rowcolor{Goldenrod}\hspace{\myspace}\textbf{Date}\hspace{\myspace} &
\hspace{\myspace}\textbf{Miles}\hspace{\myspace} &
\hspace{\myspace}\textbf{Gal.}\hspace{\myspace} &
\hspace{\myspace}\textbf{MPG}\hspace{\myspace} &
\hspace{\myspace}\textbf{Cost}\hspace{\myspace}\\
\midrule
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\ \hline
 & & & & \\
\bottomrule
\end{tabular}\\
\end{center}
```

While you will probably either use letter-size or A4-size paper, but not both, there may be times when you want a script to work on both middle and corner pages for both the Letter and A4 versions of *Plans Unfolding*. A good example of this is the Page Script Collection online, in which scripts I have uploaded will work in either version of the software on any page. In this case, an outer \ifthenelse command can be added to distinguish the software version, based on a test of whether the command \afour is defined. This becomes a bit wordy, but it allows truly universal use of a page script. Of course, determining the correct values for both letter and A4 paper in the example

below requires testing with both versions of the software, something I am willing to do for the online page scripts but something you may not feel is necessary. If you upload a page script to share in the online collection, feel free to indicate in the title or description that it only works on letter or A4 paper.

```
\ifthenelse{\isundefined{\afour}}{
  % Letter paper
  \ifthenelse{\value{ismiddlepage} = 1}{
    \setlength{\myspace}{2.8pt}  % Fit middle page
  }{
    \setlength{\ myspace}{0.6pt}  % Fit corner page
  }
}{
  % A4 paper
  \ifthenelse{\value{ismiddlepage} = 1}{
    \setlength{\ myspace}{4pt}  % Fit middle page
  }{
    \setlength{\ myspace}{2.2pt}  % Fit corner page
  }
}
```

Lines and Repetition

Let's take a look at the text script used to create the Ruler that runs across the top of Side 1 of the organizer by default. The inch ruler described here appears on letter-size output, while a metric scale appears in the A4 version. We use the BEGINDRAWING line to enter the custom drawing mode. You might think that we should use the BEGINDRAWING HORIZONTAL line since the ruler is horizontal, but we will use absolute page coordinates (see Appendix D for these absolute coordinates) rather than (PX,PY) page offsets, so we don't need horizontal rotation.

The top ruler is scaled in 16ths of an inch from the upper right corner of the side, which makes it convenient for measuring since this entire length is an accordian fold that extends very naturally for measuring.

Note that this ruler script is not entered in a Text page but is run behind the scenes by the *Plans Unfolding* software when Rulers are enabled under the General Settings menu. However, it *could* be run in a Text Page, and if so it will create the ruler regardless of the menu setting, so let's say we appended this script to the end of the existing text in, say, the title page F:

```
BEGINDRAWING
 % Using TikZ foreach rather than forloop.
 \scriptsize
 \foreach \x in {0,.03125,...,10.571} {
    \draw[OliveGreen] (10.571in -\x in,0pt) -- (10.571in-\x in,-1.5pt);
 }
 \foreach \x in {0,.0625,...,10.571} {
    \draw[OliveGreen] (10.571in -\x in,0pt) -- (10.571in-\x in,-2pt);
 }
 \foreach \x in {0,.125,...,10.571} {
    \draw[OliveGreen] (10.571in -\x in,0pt) -- (10.571in-\x in,-3pt);
 }
 \foreach \x in {0,.25,...,10.571} {
    \draw[OliveGreen] (10.571in -\x in,0pt) -- (10.571in-\x in,-5pt);
 }
 \foreach \x in {0,.5,...,10.571} {
    \draw[OliveGreen] (10.571in -\x in,0pt) -- (10.571in-\x in,-7pt);
 }
 \foreach \x in {1,2,...,10.571} {
    \draw[OliveGreen,thick] (10.571in -\x in,0pt) -- (10.571in-\x in,-10pt);
    \draw[OliveGreen] (10.571in -\x in,-8pt) node[anchor=east] {\x};
 }
```

The coordinate (0,0) is the upper-left-hand corner of the entire sheet on the organizer border, i.e., y=0 lies along the top border. The x-distance increases to the right and the y-value decreases as you proceed down the sheet. Coordinate units can be cm, in or pt (points, with 72.27 pt/inch or 2.85 pt/mm).

Any text after a % on a line is considered a comment. The TikZ \foreach command creates a for-loop where the variable \x runs from 0 to 10.571 by .013125 in the first such loop above, where here .03125 = 1/32 so the first loop is drawing the smallest tick marks every 1/32 of an inch. In this loop a \draw command creates a small green vertical line from the top border downward 1.5pt every 1/32 in from the right edge (which from the page dimensions given in Appendix D is 764pt = 10.571in from the left border). This loop is repeated for every 1/16, 1/8, ¼, ½ and 1 inch spacing, with progressively longer tick marks. In the loop for whole inches, a second \draw command creates nodes with the middle of its east side anchored at a position 8pt under the top border, and the value of the loop variable \x (the inch value) is printed as text.

The cm rulers are created in the same way (see Rulers.tex in the BuildArea folder for the LaTeX scripts for all the rulers). The protractor is drawn using individual line segments whose starting and ending coordinates were pre-computed (see Protractor.tex in the same folder). The Astronomy page is drawn from about 700 drawing commands calculated when the build is initiated, stored in the SkyMap.tex file in the BuildArea folder, and input into the main LaTeX script. (Have a look at this file after you enable the Astronomy page and run a build and you may be shocked at how fast these calculations and drawing commands actually are.)

So it may surprise you to see that from any Text page at all you can add text that draws in any other page or across pages—basically anywhere on that side of the organizer, and this might be useful for other things besides rulers. However, this is not a clean way to do things, as the text is tucked away in a local Text page. A much better way is to insert the scripts that are not page-specific (and not including a BEGINDRAWING line!) in the Side 1 or Side 2 script areas accessed from the Options→Add Custom LaTeX Code menu option.

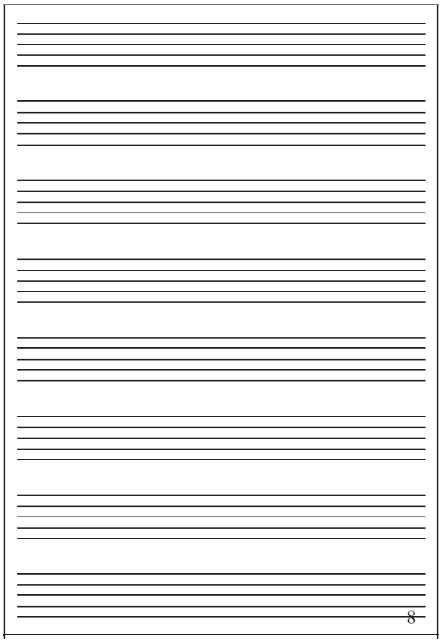
Let's look at another example of using the \foreach command in a page script. Here we will create a music staff sheet consisting of a number of rows of 5-lined staffs. This also uses the \calcprintwidth command to calculate the width of the display \printwidth based on a user-defined margin and internal knowledge of whether the page is a middle page or a narrower corner page, or for letter-size or A4 paper—in this way the script can be universal. The \calcprintwidth is only available in Plans Unfolding version 1.1 or later.

```
% ***Music Staff Sheet***
% Works only under Plans Unfolding version 1.1 or higher.
% See the Users Manual for a tutorial of this page script.
% Note: The staff color will be the color chosen for this text.

BEGINDRAWING
\providecounter{numstaffs}
\providelength{\stafflinewidth}
\providelength{\stafflinesep}
\providelength{\staffsep}
\providelength{\vertoffset}

% MODIFY DEFAULT LAYOUT PREFERENCES BELOW IF DESIRED.
\setlength{\pagemargin}{6pt} % left/right margin
\setcounter{numstaffs}{8}  % number of staffs
\setlength{\stafflinewidth}{0.5pt}  % staff line width
\setlength{\stafflinesep}{5.0pt}  % staff line separation

% DO NOT EDIT TEXT BELOW.
\calcprintwidth  % Sets \printwidth based on \pagemargin and page location
\setlength{\staffsep}{(\fullheight / \value{numstaffs}) - (\stafflinesep * 4)}
\setlength{\vertoffset}{\staffsep / 2}
\foreach \staff in {1,2,...,\value{numstaffs}} {
  \foreach \staffline in {1,2,...,5} {
    \draw[line width=\stafflinewidth](PX+\pagemargin,PY-\vertoffset) -- (PX+\pagemargin+\printwidth,PY-\vertoffset);
    \global\advance \vertoffset by \stafflinesep
  }
  \global\advance \vertoffset by -\stafflinesep
  \global\advance \vertoffset by \staffsep
}
```



The sample screenshot shows black staff lines, but as always the default drawing color is set to the color assigned to the page script text, so the blue text would produce blue staff lines. Again, any text following a % on a line is a comment. As described in the later section *Variables: Counters and Lengths* we can define integer counters or non-integer lengths with units to use as variables in LaTeX scripts. These have to be declared first if they are not pre-defined in Preamble.tex or in the user Custom LaTeX Code menu entry. The build will fail if variables are declared more than once, and we may want music staffs in multiple pages, so we use the \providecounter{} and \providelength{} commands, which will only declare them if they have not been previously declared. The value of a counter is set by \setcounter{*name*}{*newvalue*} and read as \value{*name*}. The value of a length is set by \setlength{*name*}{*value_with_units*} and read as *name*.

We could just place the staffs in a fixed layout, but here we allow the user to set variables for the left/right margin (\pagemargin, which is declared in Preamble.tex), the number of staff rows on the page (\numstaffs), the width of the staff lines (\stafflinewidth), and the distance between lines within a staff (\stafflinesep). The script then calculates the separation between staffs (\staffsep) based on the height of a page (\fullheight, defined in Preamble.tex) and these variables. It then draws the staffs with a half-separation at the top and bottom and the full separation between staffs, all centered on the page. In this way the user can run a continuous staff down two pages top to bottom (such as Page 6 and Page 7, where I once had such a continuous staff sheet) and across multiple pages left-to-right (here the user should set \pagemargin to 0pt).

First, the width of the display on the page has to be calculated based on the full page width \fullwidth (from the earlier table of variables automatically populated for each page) and the \pagemargin set here by the user. For each page the length \printwidth is always set by default for \pagemargin = 6pt, but here we allow the user to change the \pagemargin value so we need to recalculate it. As mentioned above, the \calcprintwidth command populates the \printwidth length with this value.

Now the equation for the distance between staffs is given by:

$$S = H/n - 4s$$

where S is the separation between staffs, H is the height of a page, n is the number of staffs on a page, and s is the separation between lines within a 5-line staff. You can see this formula used here to set the length of \staffsep in the script. The only nuance here is that when a length is multiplied by a constant as in "\stafflinesep * 4", the value containing the units should come before the constant or the build will fail because it assumed cm for the units of the constant and finds it multiplied by another value with units.

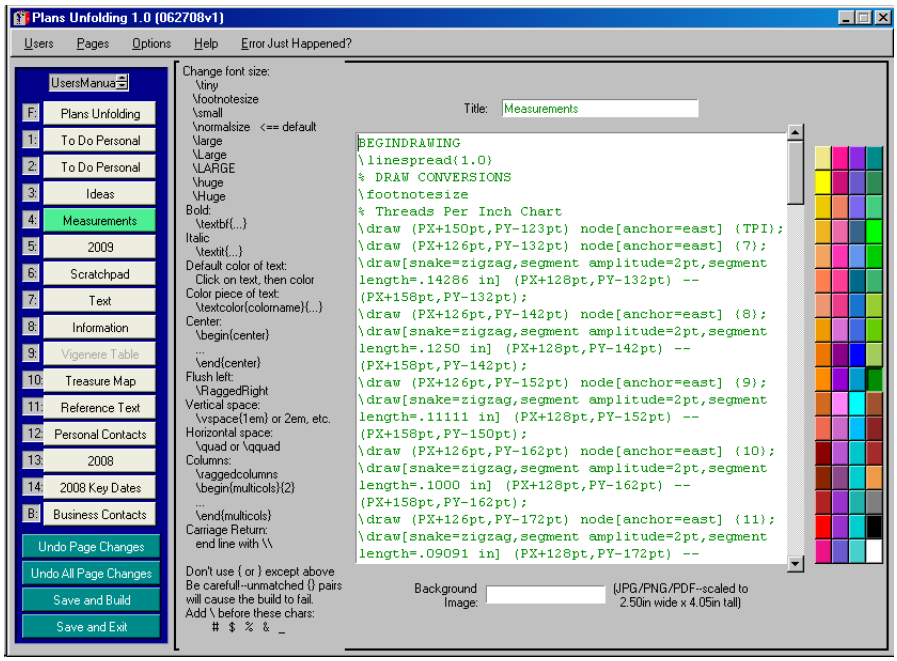
The length \vertoffset here is the offset from the top of the page for the currently drawn line (the offset to add to PY). Initially it is set to half of \staffsep. Then two nested \foreach loops are used, the outer one for the number of staffs on the page and the inner one for the 5 lines in each staff. In the inner loop a line is drawn \vertoffset from top of the page, horizontally from the left margin with a length of \printwidth. Then \vertoffset is increased by the separation between lines in a staff. In the outer loop the last \vertoffset is negated and \vertoffset is increased by the calculated separation between full staffs instead. If the default layout variables are changed to impossible values (say, 16 staffs with a line separation of 5pt), the staffs will appear unevenly spaced.

Notice that we did **not** use the command \addtolength to increase the value of \vertoffset. This is due to the odd fact that variables are by default local to a \foreach loop in TikZ, or in other words, changes to variables do not extend outside the loop. So the command \global\advance is used to add to the length, which is then a global change.

The Piano Music Staff Sheet page script in the online gallery actually uses three nested \foreach loops, the extra one for drawing the two staffs in a pair of staffs. You may want to look at this and the Guitar Tabs Staff Sheet script for additional examples.

Mixing Text Nodes and Shapes

Here's a Text page that uses both LaTeX text commands and TikZ drawing commands. This appears on **Letter-size** output as page 4 for the default user (the text area is designed to allow spaces along the edges for the protractor and rulers for when they are enabled in the General Settings menu). This page includes a table of conversion factors with approximate factors for fast estimates; a chart for finding U.S. screw and bolt sizes by placing the shaft over the box that matches its outside diameter (O.D.); and a threads-per-inch (TPI) scale to measure against screws and bolts. With the rulers this provides the part number for such a fastener, say a 8-20 1-1/2" bolt. I imagine you may want to customize some of the contents of this page for conversions you want. **Since this involves mostly non-metric units, page 4 on A4 output is instead a List page by default**.



The text for this page is quite long and complicated, so we will break it into pieces for study. We will make this for vertical pages only, which only means that we can have longer content and we can use anchors other than "north west" to simplify centering text a little bit.

The first line of the script is

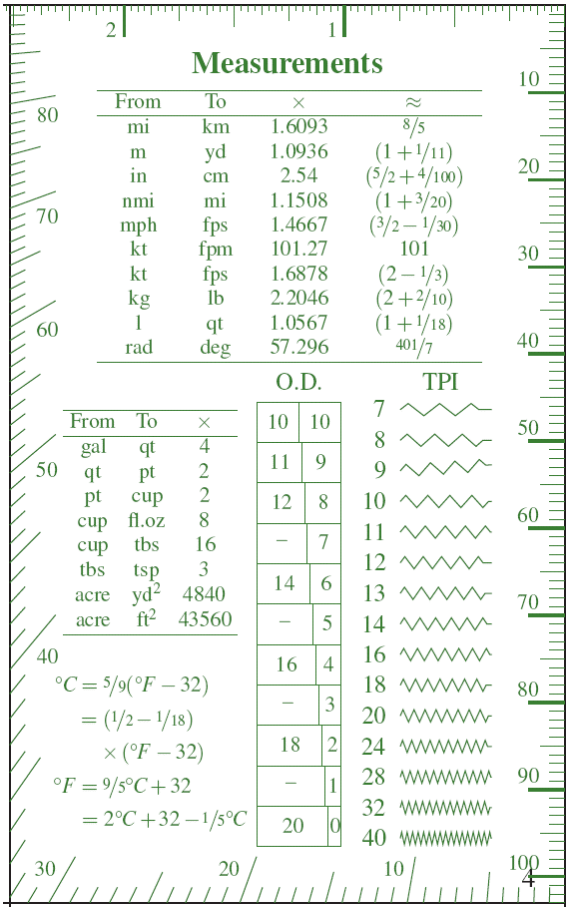
```
BEGINDRAWING
```

As in the previous examples, this line is **required** before using **any** TikZ drawing commands, because page text is actually contained within a TikZ draw node and that draw has to be terminated before a new draw command can be used. We could have had text prior to the BEGINDRAWING macro, say introductory text or a bigger title as we saw for page F, but here we will just start TikZ drawings. These will consist of nodes containing text and tables as well as vector drawings of zigzags lines.

```
\linespread{1.0}
% DRAW CONVERSIONS
\footnotesize
% Threads Per Inch Chart
```

The \linespread command followed by a font size declaration such as \footnotesize will set the line separation factor. The default in *Plans Unfolding* is 1.1, so \linespread{1.0} will bring lines closer together.

```
\draw (PX+150pt,PY-123pt) node[anchor=east] {TPI};
\draw (PX+126pt,PY-132pt) node[anchor=east] {7};
```



```
\draw[snake=zigzag,segment amplitude=2pt,segment length=.14286 in] (PX+128pt,PY-132pt) -- (PX+158pt,PY-132pt);
```

As mentioned before, PX and PY represent the horizontal x- and y-coordinates of the upper left corner of the page (before flipping upside down for upside-down pages). The x-distance increases to the right and the y-value decreases as you proceed down the page. So (PX+150pt,PY-123pt) is the coordinates of a point 150pt from the left side of the page and 123pt down from the top; again, there are 72.27 pt/inch or 2.85 pt/mm and you can also use "cm" or "in" units. The use of PX and PY macros allow custom page scripts with drawings to be pasted into any Text page.

So in the first line a "node" is drawn with its east side anchored at these coordinates on the page, and the text "TPI" is written in that node, as you can see in the output image. The text "7" is written below and to the left of it, and to the right of it a snaking, zigzag line of a given amplitude and segment length of 1/7 inches is drawn between two points. This is the top TPI scale, and text and snakes for the other scales are drawn in the same way but not shown here—you can look at the text in the default Text page 4 if you want to see them.

```
% Screw Size Chart
\draw (PX+105.5pt,PY-123pt) node[anchor=east] {O.D.};
\scriptsize
\newcounter{leftsize}
\setcounter{leftsize}{10}
\newcounter{rightsize}
\setcounter{rightsize}{10}
\draw (PX+81pt,PY-129.5pt) -- (PX+81pt,PY-274.7pt);
\draw (PX+81pt + .380in,PY-129.5pt) -- (PX+81pt + .380in,PY-274.7pt);
\foreach \x in {0,1,...,10} {
   \draw (PX+81pt,PY-129.5pt - 13.2*\x pt) -- (PX+81pt + .380in,PY-129.5pt - 13.2*\x pt);
   \draw (PX+81pt + .190in + .013*\x in,PY-129.5pt - 13.2*\x pt) -- (PX+81pt + .190in + .013*\x in,PY-129.5pt - 13.2*\x pt -
13.2pt);
   % Show only screw sizes that are officially designated.
   \ifthenelse{\value{leftsize} = 13 \OR \value{leftsize} = 15 \OR \value{leftsize} = 17 \OR \value{leftsize} = 19}{
      \draw (PX+81pt + .095in + .0066*\x in,PY-129.5pt - 13.2*\x pt - 6.5pt) node {\---};
   }{
      \draw (PX+81pt + .095in + .0066*\x in,PY-129.5pt - 13.2*\x pt - 6.5pt) node {\theleftsize}; % centers screw size by default
   }
   \draw (PX+81pt + .285in + .0066*\x in,PY-129.5pt - 13.2*\x pt - 6.5pt) node {\therightsize}; % centers number by default
   \addtocounter{leftsize}{1}
   \addtocounter{rightsize}{-1}
}
\draw (PX+81pt,PY-274.7pt) -- (PX+81pt + .380in,PY-274.7pt);
```

This is the entire text for drawing the screw outside diameter (O.D.) chart. After the "O.D" text is drawn, two new counters are created with \newcounter{} for holding integer values, leftsize and rightsize, which represent the size in the left and right cells. If you are using these counters anywhere else in your organizer, the build will fail with an error message that these counters are defined twice, so in this case these lines should be removed from the Text script and placed instead in the PREAMBLE entry accessible via the Options→Add Custom LaTeX Code menu option—then these counters are defined only once and available wherever they're needed.

The counter values are set initially to the top scale, each to 10, using the \setcounter{} command. The outside left and right vertical lines are drawn next. Then the \foreach command begins a for-loop that sequences the variable \x through the 11 rows of the chart. The horizontal and vertical line segments are drawn with offsets proportional to \x and the \ifthenelse command makes sure that only standard values of leftsize and rightsize are printed in the cells. The \addtocounter command changes the leftsize and rightsize values as the next \x row is drawn. Finally the bottom line of the scale is drawn.

```
% Conversion Tables
\draw (PX+26pt,PY-25pt) node[anchor=north west] {
\scriptsize
\begin{tabular}{cccc}
\hline
From & To & $\times$ & $\approx$\\
```

```
\hline
mi & km & 1.6093 & $\nicefrac{8}{5}$\\
m & yd & 1.0936 & $(1 + \nicefrac{1}{11})$\\
in & cm & 2.54 & $(\nicefrac{5}{2} + \nicefrac{4}{100})$\\
nmi & mi & 1.1508 & $(1 + \nicefrac{3}{20})$\\
mph & fps & 1.4667 & $(\nicefrac{3}{2} - \nicefrac{1}{30})$\\
kt & fpm & 101.27 & 101\\
kt & fps & 1.6878 & $(2 - \nicefrac{1}{3})$\\
kg & lb & 2.2046 & $(2 + \nicefrac{2}{10})$\\
l & qt & 1.0567 & $(1 + \nicefrac{1}{18})$\\
rad & deg & 57.296 & $\nicefrac{401}{7}$\\
\hline
\end{tabular}
};
\draw (PX+15pt,PY-129.5pt) node[anchor=north west] {
\scriptsize
\begin{tabular}{@{\hspace{2pt}}c@{\hspace{4pt}}c@{\hspace{4pt}}c@{\hspace{2pt}}}
\hline
From & To & $\times$\\
\hline
gal & qt & 4\\
qt & pt & 2\\
pt & cup & 2\\
cup & fl.oz & 8\\
cup & tbs & 16\\
tbs & tsp & 3\\
acre & yd$^2$ & 4840\\
acre & ft$^2$ & 43560\\
\hline
\end{tabular}
};
```

A node is drawn whose upper left (north west) corner is 26 pts right of the left side of the page and 25 pts down from the top. As we saw in an earlier example, the table is drawn with 4 centered columns but no vertical bars. Math notation in LaTeX is surrounded by $ signs, so math symbols like “times” and ≈ have special commands \times and \approx that must be surrounded by $ symbols. The \nicefrac{*numerator*}{*denominator*} command makes elegant fractions (like ½ using \nicefrac{1}{2} here)—this also has to lie within $ symbols.

A second table is started lower and to the left of the first. The @{\hspace{2pt}} adds extra space beyond that calculated for fitting the data in the columns so the table fills the available space well (we added space to the title names rather than the column definitions in our earlier examples). The ^ symbol indicates a superscript, but is a math operation so it has to lie within $ signs—here yd^2 will print out as yd^2 and yd^{23} would print out as yd^{23}. The underscore _ operates the same way but for subscripts. This may seem complicated, but in this way LaTeX forms complicated equations with sophisticated algorithms using only simple ASCII text.

```
% NOTE: Need some sort of text width defined to get CRs & aligns to work.
\draw (PX+14pt,PY-210pt) node[text width=40pt,inner sep=0pt,anchor=north west] {
\scriptsize \RaggedRight
\begin{align}
\degree C &= \nicefrac{5}{9}(\degree F - 32) \nonumber \\
&= (\nicefrac{1}{2} - \nicefrac{1}{18}) \nonumber\\
&\quad \times (\degree F - 32) \nonumber \\
\degree F &= \nicefrac{9}{5}\degree C + 32 \nonumber \\
&= 2\degree C + 32 - \nicefrac{1}{5}\degree C \nonumber
\end{align}
};
```

This last piece creates the equations for converting Fahrenheit to Celsius and back. Aligning equations in a TikZ node requires a text width of the node to be defined as an option (40pt here). Here we will be writing equations in which the = signs are automatically aligned vertically. The \begin{align} and \end{align} commands define the

align environment. Each equation is entered one per line with a carriage return \\ at the end, except that the last line has no \\ at the end. Equations are automatically numbered at the end of the line unless you add \nonumber before the carriage return. The character to be aligned in each equation must be preceded by a &, so that symbol precedes = in the equations. The \degree is the degree symbol and \quad is a small horizontal space. As always the node lies within squiggly brackets {} and the \draw command ends in ; and we are done.

Want to see what kinds of graphics can be drawn using TikZ? Have a look at the gallery of TikZ drawings (and source text) at http://www.fauskes.net/pgftikzexamples/all/ . These would have to be altered for page coordinates and widths to be used in pages here.

Variables: Counters and Lengths

We have seen in earlier examples that we can use LaTeX *counters* to hold variables. In fact, the TeX language was originally designed for straightforward typesetting only (mainly math) and it was only the efforts of later package writers that introduced the highly-useful feature of programmable typesetting with controls such as \ifthenelse{}{}{} and \foreach. They use the built-in counter feature originally used to number pages and sections to provide integer variables using the following commands:

- \newcounter{}
- \setcounter{}{}
- \addtocounter{}{}

The counter is defined with the counter name. In the last example we used the commands

```
\newcounter{leftsize}
\setcounter{leftsize}{10}
\addtocounter{leftsize}{1}
\value{leftsize}
\draw (PX+81pt + .095in + .0066*\x in,PY-129.5pt - 13.2*\x pt - 6.5pt) node {\theleftsize};
\ifthenelse{\value{leftsize} = 13 \OR \value{leftsize} = 15 \OR \value{leftsize} = 17 \OR \value{leftsize} = 19}{}{}
```

The command \value{leftsize} returns the value of the counter, while \theleftsize returns the text string of the value.

Here's a bit of text from the Preamble.tex file in the BuildArea subdirectory of the *Plans Unfolding* folder, showing how the counter can be used in a for-loop, where the \newitem{} is a newly-defined command via \newcommand{}{} found earlier in that file. Note that this \forloop{}{}{} command is a different form than the \foreach command used in earlier examples (and described in the TikZ manual)—either for-loop approach is possible. The [1] argument is the increment amount here.

```
\newcounter{numlines}
\newcounter{loopindex}
\newcommand{\fillblanklines}{
   \ifthenelse{\value{numlines}< 23}{
      \forloop[1]{loopindex}{\value{numlines}}{\value{loopindex} < 23}{
         \newitem{}
      }
   }{}
}
```

There are times, of course, when you want a non-integer variable. LaTeX supports *lengths* that can be used for this purpose. These are a bit tricky, though, because these lengths come with units such as pts, in, or by default, cm. Fortunately, it turns out we are usually interested in simple computations of coordinates and lengths here. For calculations that are more involved we should run an external program to compute values and store them into, say, a comma-separated-value (CSV) text file. These values can then be read from the file into LaTeX length variables using the *datatool* package that is automatically included by the *Plans Unfolding* application—refer to the *datatool* documentation listed in Appendix A to see how to do this.

Lengths have analogous commands to the counters:

- \newlength{}
- \setlength{}{}
- \addtolength{}{}

Here the length is defined with a \ in front, as in \newlength{\myvalue} and referred to simply by that name, as in \addtolength{\myvalue}{3.2pt} or \setlength{\myvalue}{\myvalue*4.2 + 2.3pt} or \vspace{\myvalue}. You have to add or subtract values with units, so you can have \setlength{\myvalue}{5pt*4.2 + 2.3cm} but not \setlength{\myvalue}{5pt*4.2 + 2.3}. And it happens that when multiplying by a constant you have to write the term with units first, so entering \setlength{\myvalue}{4.2*5pt + 2.3cm} can result in an error message because there are no units on 4.2. The default length unit is cm, but it's much better to always specify it.

You may want to look through the Preamble.tex file in the BuildArea directory to see more examples of these.

It turns out that if you declare a counter or length more than once, LaTeX will fail the build with a message that it already exists. **This can be a problem if the same page script is used in more than one page.** One solution is to declare them in the Preamble section under the Custom LaTeX Code menu function. However, instructions for adding these lines in that place would have to accompany any page script. A much better method is to use the following commands when defining a counter or length in a page script:

```
\providecounter{}   instead of \addcounter{}
\providelength{}    instead of \addlength{}
```

These command will only declare the variable if it hasn't been declared previously. If it has been declared, a simple warning is listed in the build output and can be ignored. These \provide… commands were used in the Music Staff Sheet page script described earlier.

Creating Custom Commands

If you have a sequence of LaTeX commands to perform a number of times, you can create your own custom command to perform it, and you can also pass variable values into the command when you call it in your script. These are analogous to subroutines in general programming. For example, there are scripts in the Page Script Gallery that plot points along scales for calculating mileage and tax/tip percentages. Here are snippets of the script for a mileage table in which a command \calcfuelxy{} is called to calculate the values of \pagex and \pagey on a page for drawing a set of tick marks:

```
\providecommand{\calcfuelxy}{}
\renewcommand{\calcfuelxy}[1]{
  % Left vertical line with upward linear scale. pagey is distance down the page.
  \pgfmathsetlength{\pagey}{\pageminy + (#1 * \fuelscale)}
  \setlength{\pagex}{\pageminx}
}

% DRAW FUEL SCALE
\color{Blue}
\foreach \fuel in {.5,1,...,\value{maxFuelInteger}} {
  \calcfuelxy{\fuel}
  \draw [line width=.5pt](PX+\pagex-2pt,PY-\pagey) -- (PX+\pagex+2pt,PY-\pagey);
}
```

We could have simply written the script in \calcfuelxy into the \foreach loop itself, but there are a few such loops to draw tick marks of different lengths, so it's more convenient to create a custom macro command. LaTeX offers the \newcommand{*name*}[*number of parameters*]{*code*} construct for this, but here we use the \providecommand

construct followed by \renewcommand so the script can be used on multiple pages, similar to the \providecounter and \providevariable constructs we saw in the previous section. Here we pass in one parameter from which x and y are calculated (this scale is a parametric plot). The TiKz math command \pgfmathsetlength{*name*}{*expression*} is used to calculate the length variables from the first passed parameter "#1". A second passed parameter would be referenced as #2 and so forth up to the number of parameters given in the brackets in \renewcommand. The \pgfmathsetlength command also supports trigonometric, logarithmic and other functions, and in fact another scale in this particular script uses sines and cosines—please refer to the Vehicle Mileage Table With Optional Mileage Calculator script online or the TiKz manual for the formats of these functions.

Creating Custom Colors

You can create a custom color name using the \definecolor{}{}{} such as

```
\definecolor{mycolor}{rgb}{0.84,0.33,0.45}
```

where here the rgb color components are mixed in the corresponding amounts 0.0-1.0. If you add this to the PREAMBLE section found under the Options→Add Custom LaTeX Code menu, this color will be available on any page in addition to the 68 default color names. Please see the xcolor package documentation at http://tug.ctan.org/tex-archive/macros/latex/contrib/xcolor/xcolor.pdf if you need more details.

Overlaying Images

You can use the TikZ drawing commands to insert a picture anywhere on a page:

```
\draw (PX+105.5pt,PY-123pt) node[text width=\fullmiddlewidth,inner sep=0pt,anchor=north west] {
   \includegraphics[width=3cm,height=0.75in]{../BackgroundImages/mypicture.jpg}
};
```

I would use \fullmiddlewidth for the text width at all times—this just sets a maximum width but the image will be scaled to the width and height given in the \includegraphics[ ]{} command. You can use any relative path to the image file—here it's located in the BackgroundImages folder alongside any background images. The image can be of .JPG, .PNG or .PDF type. Saving an image as .JPG works best for photos and images without sharp lines, .PNG works better to keep straight lines defined, and .PDF can be a high-resolution vector drawing if it is created by exporting a .EPS file from a vector drawing package.

To create a .PDF image file, you save a high-quality or vector image as an .EPS file using an application such as Adobe Illustrator or Adobe Photoshop. Then you open a command window and execute the program epstopdf.exe located in the miktex/bin directory under the directory in which you installed MiKTeX:

```
epstopdf.exe C:\MyImages\TestImage.eps
```

which will place TestImage.pdf in the epstopdf.exe directory that you can move to the image directory (or you could prepend the path to epstopdf.exe and run this command from within your image directory).

Changing Fonts

As described at http://web.image.ufl.edu/help/latex/fonts.shtml, LaTeX normally has three font families active at one time. The command \textrm changes the font to the Roman font version currently active (here Times New Roman by default), the command \textsf changes the font to the sans serif version currently active (such as Helvetica), and the command \texttt changes the font to the fixed-spaced typewriter version currently active (such as Courier).

If you open the Preamble.tex file in the BuildArea directory with a text editor, you will see that this application uses the package *mathptmx*, one of the PSNFSS suite. The PostScript fonts supported in this suite are described at http://ctan.tug.org/tex-archive/info/Free_Math_Font_Survey/survey.html and you can add a package to change this default font into the PREAMBLE section available under the Options→Add Custom LaTeX Code menu.

Table 2 at http://ctan.tug.org/tex-archive/info/Free_Math_Font_Survey/survey.html has a list of other such packages, and I have used the package *fourier* in the past to use the less formal "utopia" serif font (I much prefer serif fonts). The most comprehensive list is at http://www.tug.dk/FontCatalogue/calligraphicalfonts.html .

Here's one font I've used for script text. Enter this into the PREAMBLE section under the "Add Custom LaTeX Code" menu option:

```
\usepackage{calligra}
```

The first time you run this, MiKTeX will prompt you to download the *fundus* package that contains the *calligra* packages, then prompt you to install the *calligra* package. If you are not prompted, you will have to add these packages manually via the MiKTeX package manager as described in Appendix B.

Then you can write a piece of text with the \textcalligra{*text*} command, or use the command \calligra to change the default font to a script font.

But you may want to use a font that isn't defined by a package. There are many mysteries in this world, but few so profound, I think, as how to add TrueType fonts to a LaTeX installation. It can be done, apparently, and there is a *fontinst* tool to do it, as described in great detail in ftp://tug.ctan.org/pub/tex-archive/info/Type1fonts/fontinstallationguide/fontinstallationguide.pdf. It is considered a marked improvement over installing fonts by hand, but perhaps it is a measure of the user-friendliness of this tool that the author didn't feel compelled to complete the word "install" in its title.

Another general overview of LaTeX and fonts can be found at http://www.tug.org/fonts/. I would be interested in documenting successful attempts you might have at installing fonts that lie outside LaTeX packages, such as TrueType fonts available all over the web either as freeware or commercial products.

We have already seen the various commands for changing the font size, but here are the exact point sizes of each:

Command	Nominal Point Size	Exact Point Size
\tiny	5	5
\scriptsize	7	7
\footnotesize	8	8
\small	9	9
\normalsize	10	10
\large	12	12
\Large	14	14.40
\LARGE	18	17.28
\huge	20	20.74
\Huge	24	24.88

Sharing Page Scripts

Images and custom page scripts can be discussed in the *Plans Unfolding* forum at http://www.myreckonings.com/forum and they can be downloaded and viewed at the online Custom Page Script and Page Script Screenshot Galleries linked as child boards to this forum.

Support

The homepage of Plans Unfolding can be found at http://www.plansunfolding.com. Questions, comments and suggestions can be submitted or read on the Plans Unfolding forum at http://www.myreckonings.com/forum or emailed directly to Ron Doerfler at plansunfolding@myreckonings.com.

Submissions of background images and custom page scripts are encouraged! Please refer to the forums page at http://www.myreckonings.com/forums for links to these online galleries of user-generated content.

Copyright Notice

The *Plans Unfolding* software application and this Users Manual are ©2008, Ron Doerfler. All Rights Reserved. This program can be downloaded from the http://www.plansunfolding.com webpage but cannot be mass-distributed or incorporated into other products or collections of products without the express consent of the author. The author is not responsible for the accuracy of the astronomy page and the user must not use it for anything other than casual use. Loss of information and any effects from this application are the responsibility of the user.

Appendix A: LaTeX Add-On Packages

The following packages represent packages added to those already included in the Basic MiKTeX distribution. Online documentation for these packages is listed as well. Additional packages can be added by the user (see the *Text Type Page* section of this manual). The central repository for LaTeX packages and documentation is at http://www.ctan.org. A comprehensive list of symbols provided by various packages can be found at http://www.ctan.org/tex-archive/info/symbols/comprehensive/symbols-letter.pdf . A book such as **The LaTeX Companion** provides a single source for information on many of these and other packages.

memoir	An extensive package for creating books, etc. Includes many other packages listed below.
pgf	Includes the TikZ drawing package
colortbl	Advanced color support for table columns and rows (part of memoir but important)
xcolor	Adds foreground and background color support
multicol	Start and stop multiple columns on the same page
etex	Support for >255 registers (counters, etc.)
amsmath	AMS math package
amsfonts	AMS math fonts
amssymb	AMS math symbols
pifont	For additional symbols
graphicx	Adds options for graphics
fontenc	T1 font encoding
textcomp	Adds support for Text Companion font symbols
mathptmx	Times New Roman font as default
was	Includes gensymb for \degree, \celsius, \perthousand, \micro and \ohm
hyperref	Adds hyperlink support and PDF options
preprint	Includes fullpage used to set all margins to 1 inch and pagestyle to plain
units	Includes icefrac for fractions
calc	Infix math for modifying counters and lengths
faktor	Doesn't shrink numerator and denominator of inline fractions
microtype	Hangs punctuation to get better optical alignment and adds microkerning
colortbl	Advanced color support for table columns and rows
ragged2e	Improved justification, particularly left and right
xifthen	Provides the IF..THEN..ELSE construct.
forloop	Provides the FOR loop construct.
datatool	Read and manipulate a database from an external file
lettrine	Adds capability for dropped initial capital letters.

From the *memoir* user manual:

> The memoir class includes code either equivalent to, or extensions of, the following packages; abstract, appendix, array, booktabs, ccaption, chngcntr,chngpage, crop, dcolumn, delarray, enumerate, epigraph, framed, ifmtarg, ifpdf, index, makeidx, moreverb, needspace, newfile, nextpage, patchcmd, shortvrb, showidx, tabularx, titleref, titling, tocbibind, tocloft, verbatim, verse. The class automatically ignores any \usepackage or \RequirePackage related to these. The class also provides functions similar to those provided by the following packages, although the commands are different: fancyhdr, geometry, sidecap, subfigure, titlesec. You can use these packages if you wish, or just use the capabilities of the memoir class.

The manuals for most of these are listed below, but books such as **The LaTeX Companion** have sections that discuss the commands of most of these packages. The manuals for memoir, pgf/TikZ are simply fantastic—the memoir and colortbl manuals should be consulted for commands for tables and the pgf/TikZ manual is a must-read for drawing commands.

memoir	http://tug.ctan.org/tex-archive/macros/latex/contrib/memoir/memman.pdf
pgf/TikZ	http://tug.ctan.org/tex-archive/graphics/pgf/base/doc/generic/pgf/pgfmanual.pdf
colortbl	http://tug.ctan.org/tex-archive/macros/latex/contrib/colortbl/colortbl.pdf
xcolor	http://tug.ctan.org/tex-archive/macros/latex/contrib/xcolor/xcolor.pdf
multicol	http://tug.ctan.org/tex-archive/macros/latex/required/tools/multicol.pdf
etex	None online—see books if necessary
amsmath	http://tug.ctan.org/tex-archive/macros/latex/required/amslatex/math/amsldoc.pdf
amsfonts	http://tug.ctan.org/tex-archive/fonts/amsfonts/pdfdoc/amsfndoc.pdf
amssymb	None online—see books if necessary
pifont	???
graphicx	http://tug.ctan.org/tex-archive/macros/latex/required/graphics/grfguide.pdf
fontenc	None online—see books if necessary
textcomp	None online—see books if necessary
mathptmx	http://tug.ctan.org/tex-archive/macros/latex/required/psnfss/psnfss2e.pdf
was	(Included for gensymb) http://tug.ctan.org/tex-archive/macros/latex/contrib/was/readme.1st
hyperref	http://tug.ctan.org/tex-archive/macros/latex/contrib/hyperref/doc/manual.pdf
preprint	(Included for fullpage) http://tug.ctan.org/tex-archive/macros/latex/contrib/preprint/fullpage.pdf
units	(Includes \nicefrac) http://tug.ctan.org/tex-archive/macros/latex/contrib/units/units.pdf
calc	http://tug.ctan.org/tex-archive/macros/latex/required/tools/calc.pdf
faktor	http://tug.ctan.org/tex-archive/macros/latex/contrib/faktor/README
microtype	http://tug.ctan.org/tex-archive/macros/latex/contrib/microtype/microtype.pdf
ragged2e	http://tug.ctan.org/tex-archive/macros/latex/contrib/ms/ragged2e.pdf
xifthen	http://tug.ctan.org/tex-archive/macros/latex/contrib/xifthen/xifthen.pdf
forloop	http://tug.ctan.org/tex-archive/macros/latex/contrib/forloop/forloop.pdf .
datatool	http://tug.ctan.org/tex-archive/macros/latex/contrib/datatool/doc/datatool.pdf
lettrine	http://tug.ctan.org/tex-archive/macros/latex/contrib/lettrine/doc/lettrine.pdf
(examples)	http://tug.ctan.org/tex-archive/macros/latex/contrib/lettrine/doc/demo.pdf

Appendix B: Manual Installation of MiKTeX Packages

Unless you change the default settings, MiKTeX should detect any missing packages that are listed as required by a LaTeX script and prompt for automatic installation of them. If it does not, you may have to install the packages through the MiKTeX Package Browser found under the Start→Programs→MiKTeX 2.7 menu of your computer. Select the packages listed above, then click on "+" in the upper menu bar to install them. The repository may also have to be changed to one that contains a particular package—this is a menu option but there is a delay as the repository is synchronized. I suggest setting your repository to "United States FTP tug.ctan.org" as the most complete source—you can do this with the Change Package Repository option under the Repository menu pulldown. You can install the packages one at a time or select multiple packages at a time by holding down the Ctrl key while selecting them individually. Wait for the command window to disappear before continuing (don't exit that window). Click on Close to exit the update.

Appendix C: Updating MiKTeX and Its Added Packages

To update MiKTeX as well as any added packages to the latest version, run the MiKTeX Update Wizard found from the Start menu of your computer as Start→Programs→MiKTeX 2.7→Update. Given how long this update process takes, you probably want to wait to do this step until you try out *Plans Unfolding* and see whether you like it.

a. In the Update Source window it is acceptable to leave the default of getting the packages from a remote repository and picking a random repository—if there are problems you can choose a local repository later. If you are running under a Proxy, click on the Connection Settings button to enter than information. Click on Next to continue.

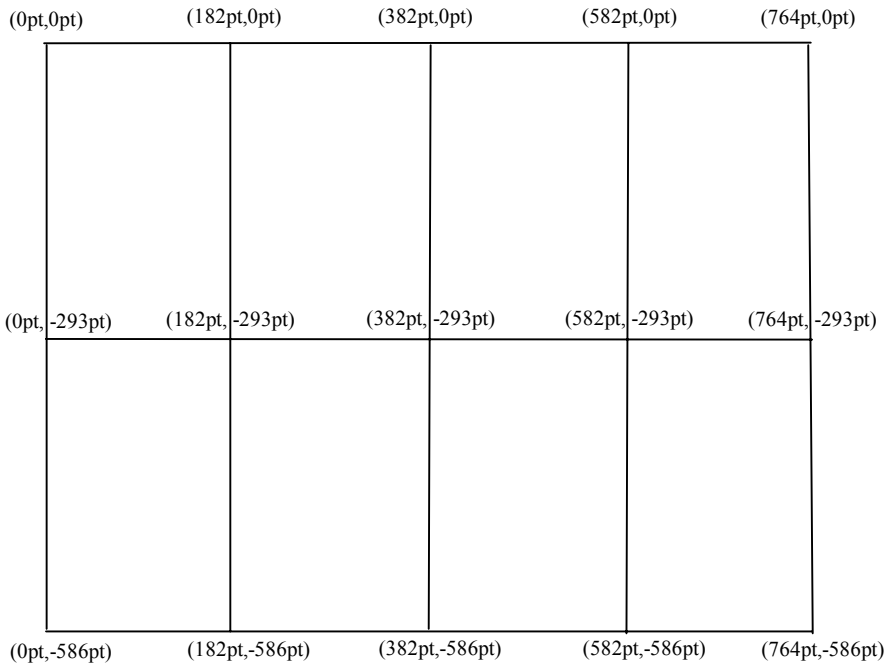
b. After the updateable packages are listed in the next window, you will probably find that the main miktex-bin file is checked and if you try to select others you will see a message that the miktex-bin file must be updated by itself first. Otherwise click on the Select All button. Then click Next when it is finally finished and Finish to exit.
c. Restart the Update Wizard if you were not able to select all the packages before. At the Updateable Packages window click on Select All and Next. Then click Next when it is finally finished and Finish to exit.

Appendix D: Absolute Page Coordinates

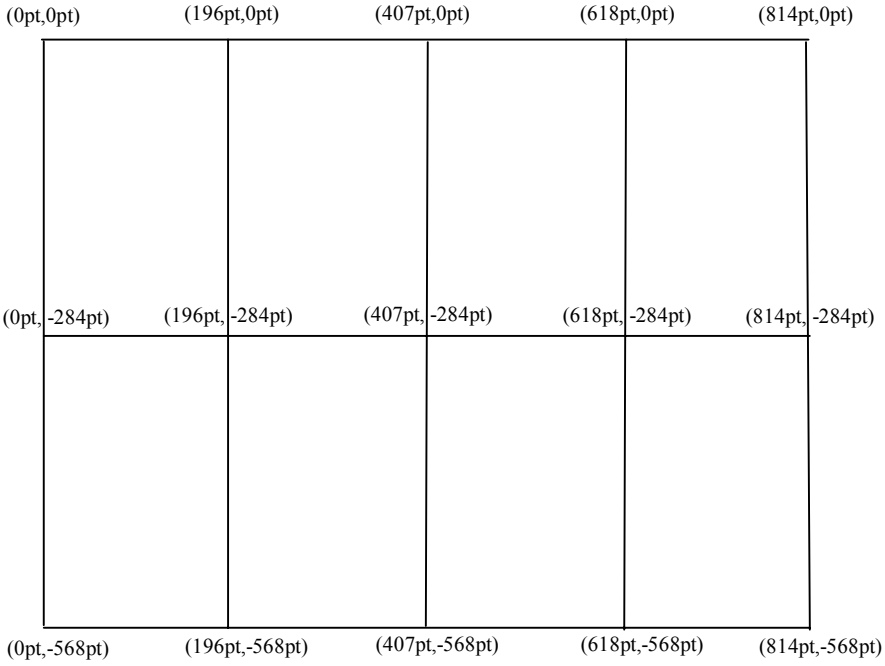
You can design custom, portable pages with text scripts that reference the default x- and y-coordinates of the upper-left-hand corner of the current page, PX and PY respectively. So if you want to place a table 10pt over and 15pt down the page, you can draw a text node at (PX+10pt,PY-15pt) as described in the *Customizing Your Organizer* section. Note that the x-coordinate increases positively as you move to the right across the page and the y-coordinate increases negatively as you go down the page. Using relative coordinates like this makes a page script usable in any page.

However, you may want to use absolute coordinates for something. These are measured relative to the upper left corner of the border on either side. It's possible to use absolute coordinates in any text page, actually, to draw anywhere on that side of the paper including in other pages, but this is not a well-structured approach—it's best to add any draw commands using absolute coordinates into either the Side 1 or Side 2 textbox under the Options→Add Custom LaTeX Code menu option. To do this you will need to know how these absolute coordinates relate to the corners of the pages, which are marked below. Notice that the 4 pages at the corners have a smaller width than the middle pages due to the borders along their sides.

Below are the absolute page coordinates for each sheet for the letter-size output. The outside edge here is the outer border of the set of pages, not the outer border of the paper itself.



Below are the absolute page coordinates for each sheet for the A4-size output:



Appendix E: Structure of the Code

It may be useful to know how the code is structured in case you want to poke around the underlying LaTeX code—maybe you're just curious or maybe you can suggest improvements that can be incorporated into the next release.

The interface is written in the Tcl-TK language and compiled to the *Plans Unfolding* executable file. Subdirectories in the directory where the application is installed include BackgroundImages (where the user places image files selected for page backgrounds) and Icons (containing images used by the interface software). The file *Settings.txt* contains all user settings as key:value strings.

When a user is added, a directory with that name is created under the main directory, and the contents of the DefaultNewUser directory are copied under it, providing a basic set of pages containing sample entries. The user directory contains the following subdirectories:

- Pages – containing the latest LaTeX script for all 16 pages, as well as a Pages.txt file that contains the page types and latest titles for the buttons.
- Backups – a copy of the Pages directory made when the interface is launched, so that page changes since the session started can be undone via the buttons in the interface.
- Calendars – containing LaTeX scripts with user entries in calendar files for the years 2008-2020.
- RecoverPages – containing copies of all Pages and Calendar files that are saved when the user saves to the Recovery Point, and copied back into the Pages and Calender directories when the user chooses to recover them.
- DebugFiles – a directory containing a collection of relevant files to be emailed to Support when the user selects to do so via the "Error Just Happened?" menu option in the interface.

In addition, the user directory contains these files:

- Preamble.tex – containing LaTeX script the user has added to the default LaTeX preamble via the Options→Add Custom LaTeX Code menu option. This might include additional packages for greater functionality (via \usepackage{} commands) or LaTeX function definitions (or macros), counters or lengths used in multiple pages.
- Side1Code.tex – containing custom LaTeX code entered by the user via the same menu function, for code that operates across all of side 1 of the organizer after the pages are drawn, such as TikZ drawing commands using coordinates with an origin at the upper left corner of the side border.
- Side2Code.tex – containing custom LaTeX code as above for side 2 of the organizer
- PlansUnfolding<*username*>.pdf – the last generated PDF printout of the organizer for this user.

When the user clicks on a button for a particular page, the LaTeX script is read from that file in the Pages subdirectory for that user and parsed into the data fields for that page type. When the user exits that page in the interface, the new LaTeX script is written back into that file and the new page is read. If the user changes the page type the new page is created from files in the Templates directory: the header is inserted that has page coordinates for that specific page number and a default script body is inserted for that page type. Special page types that are restricted to particular page numbers, such as calendar, treasure, cipher table and text, and astronomy, are copied in whole from those files in the Templates directory.

When the user selects the **Save and Build** button a number of things happen:

1. A check is made to see that all background images entered into the pages exist in the BackgroundImages directory and the user is warned if they do not.
2. The user pages are copied from the user Pages subdirectory to the BuildArea directory.
3. All custom page scripts are processed to replace page-referenced coordinates (PX and PY) with absolute coordinates for those pages (in the BuildArea copies, so the originals are still general scripts).
4. Old versions of the PDF output in the general BuildArea directory and the user directory are deleted, and if they cannot be deleted a pop-up window asks the user to exit any viewer such as Adobe Acrobat that is preventing their deletion.
5. The user preferences for side 1 and side 2 margin adjustments are entered into the files firstwidthadust.tex and seccondwidthadjust.tex that will be included in the overall LaTeX script.
6. If the user has selected the Protractor and/or Rulers to be printed, the files Protractor.tex and either Rulers.tex or LongRulers.tex (used when the protractor is not selected) are copied into the Frontaddons.tex file, all in the BuildArea directory.
7. The user-created custom LaTeX scripts in the user directory (Preamble.tex, Side1Code.tex and Side2Code.tex) are copied into the files UserPreamble.tex, Side1Code.tex and Side2Code.tex in the BuildArea directory.
8. Depending on whether symbol encoding is selected or not on the Cipher page, either VigenereAlphaSymbols.tex or VigenereAlphaOnly.tex is copied into the VigenereAlpha.tex file, all in the BuildArea directory.
9. If the Astronomy page is one of the user pages, the aa.ini file is loaded with latitude/longitude information and the aa.exe file is launched to create ephemerides for the planets, sun and moon as specified in the aain.txt file. These files are all in the main directory. That program populates this date into the aaout.txt file in the BuildArea, at which time the interface code parses this text as well as star positions and magnitudes from the myyale.csv file, constellation line positions from the MyConlines.csv file, and constellation names and positions from the MyCnames.csv file. It calculates offsets based on the user-entered location and timezone and finally produces several hundred lines of TikZ drawing commands in the SkyMap.tex file located in the BuildArea directory. This all takes a couple of seconds.
10. The pdflatex.exe file of MiKTeX is launched with a target of PlansUnfolding.tex and with output results written to the file buildoutput.tex , both in the BuildArea directory.
11. When the build is done, the build output is copied into the Build Results text window in the interface for debugging. The interface scans the output for fatal errors and checks to make sure that the output PDF file has been created. If it reads an error or no output file exists, the interface reports a build failure, otherwise it copies the PDF file to the user directory and launches it for viewing.

    The PlansUnfolding.tex target file contains the following (non-comment) text:

```
\documentclass[10pt,landscape,oneside]{memoir}   Starts a memoir-class document
\include{Preamble}                               Includes the default Preamble.tex script
\input{UserPreamble.tex}                         Includes any custom user preamble

\begin{document}                                 Ends the preamble and starts document code
\input{firstwidthadjust.tex}                     Includes side 1 user-specific margin adustments
%*****FRONT SIDE*****
\begin{tikzpicture}                              Begins a TikZ picture covering all of side 1
\setcounter{pagenum}{0}                          Sets counter variable to page number for possible script use
\setcounter{invertedpage}{1}                     Sets counter variable to indicate page is upside-down
\edgepagesetup                                   Sets page script variables for widths of edge (corner) page
\input{P0.tex}                                   Inputs user page 0 ("F") script file
\setcounter{pagenum}{1}                          Sets counter variable to page number for possible script use
\setcounter{invertedpage}{0}                     Sets counter variable to indicate page is right-side up
\edgepagesetup                                   Sets page script variables for widths of edge (corner) page
\input{P1.tex}                                   Inputs user page 1 script file
\setcounter{pagenum}{2}                          Sets counter variable to page number for possible script use
\setcounter{invertedpage}{0}                     Sets counter variable to indicate page is right-side up
\middlepagesetup                                 Sets page script variables for widths of middle page
\input{P2.tex}                                   Inputs user page 2 script file
\setcounter{pagenum}{3}                          Sets counter variable to page number for possible script use
\setcounter{invertedpage}{0}                     Sets counter variable to indicate page is right-side up
\middlepagesetup                                 Sets page script variables for widths of middle page
\input{P3.tex}                                   Inputs user page 3 script file
\setcounter{pagenum}{4}                          Sets counter variable to page number for possible script use
\setcounter{invertedpage}{0}                     Sets counter variable to indicate page is right-side up
\edgepagesetup                                   Sets page script variables for widths of edge (corner) page
\input{P4.tex}                                   Inputs user page 4 script file
\setcounter{pagenum}{13}                         Sets counter variable to page number for possible script use
\setcounter{invertedpage}{1}                     Sets counter variable to indicate page is upside-down
\edgepagesetup                                   Sets page script variables for widths of edge (corner) page
\input{P13.tex}                                  Inputs user page 13 script file
\setcounter{pagenum}{14}                         Sets counter variable to page number for possible script use
\setcounter{invertedpage}{1}                     Sets counter variable to indicate page is upside-down
\middlepagesetup                                 Sets page script variables for widths of middle page
\input{P14.tex}                                  Inputs user page 14 script file
\setcounter{pagenum}{15}                         Sets counter variable to page number for possible script use
\setcounter{invertedpage}{1}                     Sets counter variable to indicate page is upside-down
\middlepagesetup                                 Sets page script variables for widths of middle page
\input{P15.tex}                                  Inputs user page 15 script file
\frontpagenumbers                                Macro defined in Preamble.tex to draw page numbers
\drawfoldlines                                   Macro defined in Preamble.tex to draw foldlines
\input{Frontaddons.tex}                          Includes Protractor and/or Ruler scripts
\input{UserSide1.tex}                            Includes any custom user scripts for side 1
\end{tikzpicture}                                Ends side 1 TikZ picture
\end{adjustwidth}                                Ends side 1 margin adjustments
\clearpage % go to next page                     Goes to second page of ouput

%*****BACK SIDE*****                             Repeats above for side 2
\input{secondwidthadjust.tex}
\begin{tikzpicture}
\setcounter{pagenum}{12}
\setcounter{invertedpage}{1}
\edgepagesetup
\input{P12.tex}
\setcounter{pagenum}{5}
\setcounter{invertedpage}{0}
\edgepagesetup
\input{P5.tex}
\setcounter{pagenum}{8}
\setcounter{invertedpage}{0}
```

```
\middlepagesetup
\input{P8.tex}
\setcounter{pagenum}{9}
\setcounter{invertedpage}{0}
\middlepagesetup
\input{P9.tex}
\setcounter{pagenum}{7}
\setcounter{invertedpage}{1}
\edgepagesetup
\input{P7.tex}
\setcounter{pagenum}{6}
\setcounter{invertedpage}{1}
\edgepagesetup
\input{P6.tex}
\setcounter{pagenum}{10}
\setcounter{invertedpage}{1}
\middlepagesetup
\input{P10.tex}
\setcounter{pagenum}{1}
\setcounter{invertedpage}{1}
\middlepagesetup
\input{P11.tex}
\backpagenumbers
\drawfoldlines
\input{Backaddons.tex}
\input{UserSide2.tex}
\end{tikzpicture}
\end{adjustwidth}
\end{document}                                   Ends memoir-class document
```

The \edgepagesetup and \middlepagesetup commands that precede each \input{Px.tex} line provide a set of variables (a counter and five lengths) that can be used by page scripts if desired—we saw the use of these in the section on writing page scripts that work on both corner and middle pages. For example, the \edgepagesetup command is defined in Preamble.tex as:

```
\setcounter{ismiddlepage}{0}                    Gives page location (corner =0, middle=1) for scripts
\setlength{\fullwidth}{\fulledgewidth}          Sets full width of page for scripts (a corner page here)
\setlength{\pagewidth}{\edgewidth}              Sets width of page with small indent for scripts (a corner page here)
\setlength{\indentedwidth}{\indentededgewidth}  Sets width of page with double indent for scripts (a corner page here)
\setlength{\pagemargin}{6pt }                   Default page margin for page script (for \printwidth = \pagewidth)
\setlength{\printwidth}{\pagewidth}             Default display width for page script (=6pt margin)
```

A command \calcprintwidth (defined in Preamble.tex) can be called to recalculate the \printwidth length if the user can change the value of \pagemargin from its default value of 6pt. It simply sets \printwidth to the current \fullwidth value minus twice the \pagemargin value.

A page script can just use any of these lengths it wants as the page width available to use in a TikZ text node or in other drawing calculations. It can also use the \ifthenelse{}{}{} command to change the width of columns of tables, say, depending on \value{ismiddlepage}. For example, in the script piece below, \columnpadding might be a bit of extra space added before and after column headings in a table as seen earlier in the Gas Mileage table example, where we want less padding for a narrower edge page. This allows us to use the same page script regardless of whether it is used in a corner or middle page for a given paper size.

```
\providecounter{columnpadding}
\ifthenelse{\value{ismiddlepage} = 0}{
   \setlength{\columnpadding}{.5pt}
}{
   \setlength{\columnpadding}{.07cm}
}
```

If the user edits the value of a predefined variable called \pagemargin to 0pt for the full page width, 6pt for a typical indented page width, or 12pt for a doubly indented page width, or to any other value, then the page script can call a function \setprintwidth (defined in Preamble.tex) to set the length \printwidth to the appropriate value for the page margin and page location—we saw this in the Music Staff example earlier in this manual. As noted above, this length \printwidth is set by default for a page margin of 6pt on the current page.

The dummy command \afour is defined in the Preamble.tex file for only the A4 version of *Plans Unfolding*. In this way the user can provide different LaTeX script for a letter-size output than for an A4-size output using the construct \ifthenelse{\isundefined{\afour}}{*Letter-specific code*}{*A4-specific code*). We saw this earlier in our discussion of writing page script that work on any page and for either paper size.

The Preamble.tex file in the BuildArea contains scripts for all the added packages, the center-top coordinates of every page, and a number of macros for creating entries in the list-type pages, etc., that are used in the individual pages. It also includes the values of the lengths used above. You may want to look at this file.

A typical page file, one for a List-type page 1, is shown below:

```
\draw \xya node[text width=\fulledgewidth,inner sep=0pt,anchor=north, rotate=0] {
%\includegraphics[width=\fulledgewidth,height=\fullheight]{../BackgroundImages/}
};
\draw \xya node[text width=\edgewidth,inner sep=0pt,anchor=north, rotate=0] {
\vspace{8pt}
\pagetitle[Black]{To Do Personal}
%  Format: \newitem[color]{text} with [color] optional, 20 max w/o reducing font.
\newitem[Red]{Call phone company about DSL line}
\newitem[Blue]{Return book to Jeff}
\fillblanklines
};
\normalsize
% PageSettings:Version110:trailsetyes:
```

The first TikZ \draw command creates the selected background image on the pages—here there was none entered into the interface field so the line is commented out with a % symbol. It is rotated 0 degrees because it is an right-side-up page—otherwise it would be 180 degrees. The macro \xya is defined in Preamble.tex as the coordinates of the top center of page 1, so with the node anchored there a rotation of 180 degrees will just flip the image upside-down on the page.

The second \draw command is the main text node that overlays the background image. The \pagetitle command is a general macro defined in the Preamble.tex file. Each list item entered by the user has another macro \newitem to enter it in the proper position if the trailing line is checked on the interface (as indicated by the trailsetyes page setting on the last line), otherwise a \newitemnotrail macro is called. These macros increment a counter for the number of entries, so the \fillblanklines macro in that file draws empty lines for all remaining entries of the 23 possible entries. The "};" line ends the current draw command and the \normalsize line resets the default font to that size. The PageSettings comment line contains page-specific information such as the version number of the file format (useful if changes are needed in future features) and the user setting for trailing lines.

Each page type has its own format like this, and if you are interested you can view them with any text editor. If the editor sees the BEGINDRAWING line it ends the default text node with a "};" line and allows the user to create their own drawing commands for custom designs. It also adds rotations and anchors for vertical pages, or for horizontal pages if HORIZONTAL appears on the BEGINDRAWING line.

Appendix F: Troubleshooting

Q: I get no response from the Plans Unfolding application when I try to build the application for the first time.

a. Have you installed MiKTeX? If not, follow the download and installation instructions.

b. Is "pdflatex.exe" found on your computer? MiKTeX adds the path to the file when it is installed, but perhaps administrative permissions prevented it. Or, as has happened, a System Restore may have been performed on your computer that restored the system to a time before MiKTeX was installed. Open a Command Prompt window from the Start→Programs→Accessories menu. Type "pdflatex.exe". If the system responds that this is not a recognized command, then you should delete and re-install MiKTeX from scratch as described in the Installation instructions. Simply adding the folder containing *pdflatex.exe* to your PATH system variable is not sufficient, as there are other registry entries created when MiKTeX is installed that will require a re-install to create.

c. There may be administrative restrictions on your login account. Try an administrator account if you are not sure.

Q: The application hangs when it launches.

a. There may be LaTeX script in one of your Text pages that caused invalid text to be saved to your page when it was stored previously. If the interface hangs, you cannot recover your pages from the application itself as the menus are not available, so exit the application using the Windows Task Manager (called via Ctrl-Alt-Delete) and run the ***Recover_Pages_If_Application_Hangs.exe*** file located in the *Plans Unfolding* installation directory. Enter your user name and click OK, then restart the main *Plans Unfolding* application to return to the last configuration saved to the Recovery Point, and resume from there.

Q: The printer cuts off some page content near the borders.

a. Some printers simply won't print to within 1/4" of the edge of the paper where the organizer border lies. If this cuts off the border and a bit of blank space in the edge pages, this is not a big deal. However, if you are losing printed content on the sides, and you have already tried the printer settings above that do **not** re-scale or auto-center the output, then there is a way to print out the whole organizer but just slightly smaller in size. Please see the section *Printing and Folding When There Are Large Printer Margins* for instructions for this situation.

For answers to other Troubleshooting questions, please see the *Support* section of this manual.